INTERFACE EM LINGUA NATURAL PARA BANCOS AGROPECUÁRIOS DO IBGE

Vania Costa Telles

Tese submetida ao corpo docente da Coordenação dos Programas de Pós-Graduação em Engenharia da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências em Engenharia de Sistemas e Computação.

Aprovada por:

Profª. Sueli Bandeira Teixeira Mendes, Ph. D.
(presidente)

Profª. Leila Maria Ripoll Eirizik, D. Sc.

Profª. Rosana de Saldanha da Gama Lanzelotte, D. Sc.

Rio de Janeiro, RJ - BRASIL
Abril, 1994

Telles, Vania Costa.
AGRON: Uma Interface Baseada em Linguagem Natural para Consulta a Bancos de Dados da Produção Agrícola

Inclui bibliografia e índice.
132 p. 29,7 cm (COPPE/UFRJ, M. Sc., Engenharia de Sistemas e Computação, Rio de Janeiro, 1994)
Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Inteligência Artificial - Processamento de Linguagem Natural. 2. Linguística Computacional. 3. Bancos de Dados - Interfaces em Linguagem Natural.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para obtenção do grau de Mestre em Ciências (M. Sc.).

AGRON: UMA INTERFACE BASEADA EM LINGUAGEM NATURAL PARA CONSULTA A BANCOS DE DADOS DA PRODUÇÃO AGRÍCOLA

Vania Costa Telles
Abril, 1994

Orientadora: Profª. Sueli Bandeira Teixeira Mendes
Programa: Engenharia de Sistemas e Computação

Este trabalho descreve uma interface para acesso a bancos de dados que utiliza uma gramática semântica de casos para processar questões em Linguagem Natural sobre a produção agrícola no Brasil.

O protótipo implementado da interface AGRON consiste num analisador direcionado por 70 regras gramaticais que acessa um dicionário de cerca de 350 palavras para traduzir a sentença em linguagem natural num comando correspondente da linguagem de *query* SQL.

O algoritmo de análise é dirigido pela semântica, ou seja, é guiado por uma estrutura de casos semânticos objeto-espaço-tempo-restrição e pelo significado dos construtos da linguagem em termos das relações do banco de dados alvo.

Uma característica importante é o processamento de enunciados elípticos segundo uma taxonomia de elipses no diálogo, o que simplifica a interação do usuário com o sistema e aumenta o poder expressivo da interface.

Embora a versão implementada seja orientada ao domínio, é possível a adaptação ou incorporação de novos domínios semânticos espaço-temporais sem grande esforço, acrescentando-se as entradas léxicas correspondentes e certas regras gramaticais para construções inéditas.

Além disso, propomos uma extensão transportável da interface, onde a aquisição de semântica se dá numa forma interativa, e a tradução para uma forma lógica intermediária reflete o esquema conceitual do banco de dados.

O desempenho alcançado pelo sistema se mostrou bastante satisfatório dentro do objetivo principal de processar comandos imperativos e perguntas sobre a base de dados da produção agrícola no IBGE (Instituto Brasileiro de Geografia e Estatística).

Abstract of Thesis presented to COPPE/URFJ as partial fulfillment of the requirements for the degree of Master of Science (M. Sc.).

AGRON: A NATURAL LANGUAGE INTERFACE FOR QUERYING DATA BASES OF AGRICULTURE PRODUCTION

Vania Costa Telles
April, 1994

Thesis Supervisor: Sueli Bandeira Teixeira Mendes
Department: Computation and System Engineering

This work describes a data base interface which uses a case semantic grammar to process Natural Language queries about agriculture production in Brazil.

The implemented prototype of AGRON system consists of a parser directed by 70 grammatical rules that uses a dictionary including around 350 words to map the natural language sentence in a statement of SQL query language.

The parsing algorithm is semantically-driven, which means that is it guided by a object-space-time-restriction semantic case frame, and by the sense of language constructs in terms of data base relations.

An important feature is elliptical utterance processing using a taxonomy of dialogue ellipsis which simplifies the interaction between user and system, and augments the expressive power of the interface.

Although the implemented version is domain-oriented, it is possible to adapt or incorporate other space-time semantic domains with little effort, by adding the corresponding lexical entries and some grammatical rules for new constructions.

Furthermore, we propose a transportable extension of the interface, where semantic acquisition is made in a interactive manner, and the traduction in a intermediate logical form reflects the conceptual data base schema.

The performance achieved by the interface is quite satisfying according the main objective of processing imperative statements and questions over the agriculture production data base in IBGE (Brazilian Institute of Geography and Statistics).

Índice

I. INTRODUÇÃO

"... se houvessem [máquinas] que tivessem a semelhança de nossos corpos, e imitassem nossas ações tanto quanto fosse moralmente possível, disporíamos sempre de dois meios certíssimos de reconhecer que elas eram por isso verdadeiros homens. O primeiro consiste em que essas máquinas nunca poderiam usar de palavras nem de outros sinais, combinando-os, como nós fazemos para declarar aos outros nossos pensamentos (...) O segundo meio é que, conquanto as referidas máquinas fizessem muitas coisas tão bem ou melhor do que nós, elas falhariam indubitavelmente em algumas outras por onde descobriríamos que elas não agiam com conhecimento (...): donde resulta que é moralmente impossível que haja numa máquina arranjos tão diversos a ponto de fazê-la agir, em todas as ocorrências da vida, do mesmo modo por que a nossa razão nos faz agir."

R. Déscartes, *O Discurso do Método*

Desde a revolução científica do século XVII, Déscartes já havia lançado o megadesafio da Inteligência Artificial: desenvolver uma máquina tão sofisticada que simulasse a inteligência, a capacidade linguística e o comportamento humanos, a ponto de ser confudida com o próprio homem. Certamente estamos ainda muito distantes de tal objetivo, embora os cientistas houvessem previsto um progresso neste sentido muito mais rápido do que realmente vem ocorrendo.

A dificuldade em se definir modelos computacionais completos é notada, especialmente, nos ramos da Inteligência Artificial que pesquisam a capacidade linguística do ser humano, dada a complexidade que envolve a utilização de uma língua como forma de expressão. O uso de uma linguagem foi um passo tão importante na história do homem que muitos antropólogos afirmam que esta é, de fato, a característica primordial que distingue o homem dos outros animais.

A primeira aplicação prática no uso do computador para tratamento da Linguagem Natural (LN) humana escrita foi a tradução automática de uma língua para outra. Atualmente, esta aplicação interessa, particularmente, ao Mercado Comum Europeu, que precisa lidar com documentos escritos em mais de sete línguas oficiais! Apesar dos esforços, até agora nenhum sistema de tradução automática por computador foi desenvolvido, somente tradução *auxiliada* por computador, cujo resultado precisa ser aprimorado por um especialista.

De fato, até o presente momento, não houve ninguém que desenvolvesse um sistema completo de compreensão da linguagem natural, que possa lidar de maneira consistente e eficaz, por exemplo, com as mudanças de contexto subliminares em um discurso, com os aspectos intencionais dos atos de fala, enfim, com atributos

psicolinguísticos humanos complexos que envolvem o senso de humor, a intuição, e a imaginação, entre outros. O que existe são pesquisadores que trabalham com subconjuntos da linguagem natural, na tentativa de modelar sistemas que possam processar estes subconjuntos, e que simulem e/ou auxiliem, até certo ponto, a execução de tarefas que exijam inteligência.

Assim, surgiram as áreas de Processamento de Linguagem Natural e Linguística Computacional, dois subramos da Inteligência Artificial que tratam mais comumente da LN escrita, e que se confundem, porém com um objetivo comum: criar modelos computacionais para compreensão da Linguagem Natural. Devido a amplitude e a interdisciplinaridade da pesquisa, estas duas áreas estão intimamente relacionadas a outros campos da ciência tais como a linguística, a psicologia, a filosofia e a ciência da computação propriamente dita.

O objetivo maior dos modelos computacionais para processamento de Linguagem Natural é especificar uma teoria que possa compreender e gerar uma linguagem com tanta fluência quanto um orador humano, entendendo-se por fluência a variedade de conhecimentos, de construções linguísticas, e os raciocínios adequados para lidar com estes conhecimentos. Para tanto, existem duas motivações básicas.

Primeiramente, temos a motivação tecnológica, que se resume em construir sistemas computacionais inteligentes para facilitar a comunicação homem-máquina, tais como interfaces em Linguagem Natural para acesso a banco de dados, sistemas de tradução automática, sistemas de análise de texto, sistemas de compreensão da fala, entre outros.

Em segundo lugar, está a motivação linguística, ou da ciência cognitiva, que é compreender melhor os mecanismos pelos quais os homens se comunicam utilizando a Linguagem Natural. Esta segunda motivação é dividida com os linguistas, neurolinguistas, psicólogos, filósofos, e outros teóricos.

A compreensão e o uso da linguagem estão ligados ao conhecimento sobre a estrutura da linguagem em si, o discernimento sobre as palavras, como combiná-las em sentenças, quais os seus significados, como estes significados contribuem para o significado global de um enunciado, enfim como utilizar os recursos da língua de forma correta e coerente. O conhecimento da linguagem envolve os estudos de lexicologia, morfologia, fonética, ortografia, gramática, além da semântica de palavras e frases.

Outro aspecto fundamental é o conhecimento sobre o que faz o homem inteligente - sua visão geral do mundo e sua capacidade de raciocínio - a ponto de fazê-lo capaz, por exemplo, de responder perguntas e participar de uma conversa.

Digamos, qualquer pessoa *sabe* que a sentença "Luzes tristes têm pernas compridas" não faz sentido, embora seja correta do ponto de vista gramatical. Este aspecto envolve os chamados conhecimentos contextual, pragmático, e conhecimento do mundo.

Os sistemas de processamento de LN podem, geralmente, ser divididos em três etapas mais ou menos distintas: a primeira etapa realiza uma análise léxica/gramatical das sentenças; a segunda, uma análise semântica; e finalmente, segue-se uma análise contextual. Esta última etapa pode ser especialmente trabalhosa pois exige a utilização de técnicas elaboradas de representação do conhecimento, incluindo, eventualmente, o uso de modelos de estrutura de discurso, modelos de crenças e de atos de fala.

Nesta dissertação, estaremos concentrados na análise e desenvolvimento de Interfaces em Linguagem Natural (ILN) para consulta a bancos de dados. Estas interfaces tratam, principalmente, da análise gramatical e semântica das sentenças, uma vez que o banco de dados, pela própria estrutura das informações nele armazenadas, define um contexto específico. Isto é, se um banco de dados é sobre amostras lunares, este é justamente o contexto da aplicação que faz, por exemplo, com que a palavra *missão* se refira a uma missão lunar, e não a uma missão militar. Em muitos casos, algum conhecimento pragmático também precisa ser embutido na interface.

É importante observar que existem trabalhos bem sucedidos no campo das interfaces ditas *transportáveis*, ou seja, que se adaptam a diversos domínios de aplicação, incorporando um conhecimento semântico e contextual mais amplos. Um bom exemplo de interface transportável é a interface TEAM [Grosz et al, 1985], desenvolvida no Centro de Inteligência Artificial da Califórnia, EUA, que será discutida posteriormente.

Atualmente, são inúmeras as interfaces para bancos de dados desenvolvidas para os diversos idiomas, embora muito poucas voltadas para o português. Certamente, este é o tipo de tecnologia que não podemos importar, simplesmente porque depende, essencialmente, das características da própria língua. As dificuldades no desenvolvimento de técnicas e sistemas nacionais utilizando o idioma português reflete, por um lado, a falta de incentivos para a pesquisa em geral, e por outro, o desestímulo dos próprios estudantes e profissionais, devido a complexidade do assunto ou aos resultados infrutuosos e lentos que têm acompanhado o estudo da Linguagem Natural.

Neste trabalho propomos a criação da interface AGRON, uma interface baseada em LN para acesso a banco de dados da Produção Agrícola no Brasil, utilizando informações obtidas através das pesquisas do IBGE (Instituto Brasileiro de

Geografia e Estatística). O projeto é baseado numa necessidade emergencial real de disseminar de forma transparente e ampla, para usuários inexperientes, toda a riqueza de dados que integra o acervo do IBGE. Um sistema que utilize uma entrada em LN têm se mostrado bastante indicado para atender tal objetivo.

A interface AGRON é voltada para a consulta a bancos de dados com um caráter local-histórico, isto é, com informações sobre um determinado objeto e suas variações numa dimensão espaço-tempo. O subconjunto do português processado pelo AGRON é baseado numa gramática semântica que utiliza uma estrutura de *casos* para definir o arcabouço espaço-temporal que caracteriza a interface. Além disso, o tratamento de sentenças elípticas é visto como um recurso importante, usado para simplificar a interação usuário-máquina e como ferramenta que aproxima o sistema do diálogo humano.

Na implementação de um protótipo da interface AGRON, utilizamos o banco de dados das pesquisas agrícolas do IBGE, esperando que, futuramente, o AGRON possa ser implantado na sua totalidade, e ampliado a fim de comtemplar outros domínios de aplicação. A incorporação de certos domínios é imediata, como é o caso do domínio das pesquisas pecuárias, de outros, porém, é mais trabalhosa. A implementação atual foi feita em PASCAL, porém a filosofia geral da interface independe da linguagem de programação. A entrada do programa é uma sentença em Linguagem Natural, e a saída é um comando correpondente na linguagem de *query* SQL para acesso ao banco de dados em questão.

A área de Interfaces em Linguagem Natural tem despertado interesse crescente nos pesquisadores da Inteligência Artificial. Possivelmente, este interesse se deve a dois fatos: um sistema computacional que interage em LN é uma forma atraente e natural para se estabelecer a comunicação homem-máquina, notadamente quando se deseja obter dados de um banco de dados; além disso, a possibilidade de dotar os computadores com tal habilidade torna-os mais amigáveis. No futuro, espera-se que um usuário inexperiente possa ter contato com um ambiente computacional e obter os serviços desejados sem necessidade de conhecimentos específicos ou de treinamento prévio.

II. INTERFACES EM LINGUAGEM NATURAL

Interfaces para Acesso a Bancos de Dados

Tipicamente, os bancos de dados armazenam grande quantidade de dados que tem de ser estruturados de uma forma tal que garanta um acesso eficiente para atender o maior número possível de questões. Neste sentido, muitas linguagens de *query* têm sido projetadas para simplificar o problema de extrair dados destes repositórios. Estas linguagens podem ser linguagens artificiais de uma dimensão - compostas de letras, números e sinais matemáticos - ou linguagens gráficas de duas dimensões, que permitem ao usuário a manipulação de diagramas, ícones e menus numa série de telas.

Embora algumas linguagens de *query* possam ser atraentes e aparentemente práticas, as mesmas críticas se aplicam seja qual for a forma de apresentação: para formar uma *query* o usuário precisa não só de uma descrição da estrutura do banco de dados como também do conhecimento de uma sintaxe artificial, além de elementos de navegação e outros recursos pouco naturais necessários a elaboração dos comandos.

Um outro problema é que a completude numa linguagem formal de *query* desnecessariamente complica a definição da linguagem para os usuários inexperientes. A ênfase numa sintaxe completa e em regras de precedência para combinar operadores lógicos podem complicar superfluamente a forma lógica, além de não permitir certas construções de *queries* mais elaboradas. Por exemplo, ao invés de especificar uma sintaxe adicional para eliminar ambiguidades em situações pouco comuns, talvez seja mais interessante simplesmente apresentar as múltiplas interpretações ao usuário quando elas surgirem, e deixar que ele decida pela opção adequada.

Além disso, a formulação de uma *query* pode ser muito mais complexa do que se poderia esperar a partir da sentença correspondente em Linguagem Natural. Na prática, a complexidade destas linguagens costuma frustrar os usuários inexperientes. A pesquisa na área de interfaces em LN para bancos de dados busca eliminar esta complexidade, livrando os usuários da necessidade de um exato conhecimento sobre a estrutura dos dados e de aprender linguagens de *query*.

Poder-se-ia argumentar que a utilização de linguagens baseadas em interfaces gráficas simplificam o problema de ter que aprender uma sintaxe para a linguagem. Neste caso, o usuário se orienta apenas por ícones, menus ou telas auto-explicativas. Por outro lado, este recurso se mostra ineficiente em outras situações, conforme veremos nos resultados apresentados no estudo de Bell & Rowe que discutimos detalhadamente no último capítulo.

As interfaces em LN para consulta a bancos de dados permitem que os usuários recuperem informações através de perguntas ou comandos que são formulados, até certo ponto, da maneira que o usuário pensa normalmente, independente da forma como o dado está efetivamente armazenado. Obviamente, quando se usa o termo "Linguagem Natural" espera-se que o subconjunto da linguagem utilizado seja suficientemente "denso" para permitir que o usuário não necessite traduzir, conscientemente, as suas perguntas em termos apropriados da interface.

Assim sendo, o usuário não depende do conhecimento de uma linguagem de *query*, nem fica restrito às limitações impostas por um sistema de telas, menus e ícones. Ele usa apenas o conhecimento que já tem da sua língua pátria, que certamente é sempre muito maior do que o conhecimento que possa estar embutido numa interface. Neste sentido, o fardo do aprendizado é transferido do usuário para o sistema, ou seja, são as interfaces é que precisam se adaptar e "aprender" cada vez mais sobre a linguagem, e não o inverso.

Outros pontos positivos nas ILNs para *query* a bancos de dados serão analisados nos resultados finais. Notadamente, observamos que a Linguagem Natural é vista como uma linguagem conveniente para consultar bancos de dados, pois estes geralmente apresentam uma visão de contexto definida e simplificada, facilitando o desenvolvimento das interfaces. Embora muito esforço tenha sido aplicado no sentido de projetar interfaces cada vez mais completas linguisticamente, é importante lembrar que, na verdade, o que existe hoje são interfaces *baseadas* em Linguagem Natural, visto que os sistemas utilizam sempre um subconjunto da língua em questão.

No desenvolvimento de ILNs, defendemos a idéia de é preciso trabalhar menos com a sintaxe e mais com a semântica para alcançar resultados mais satisfatórios do ponto de vista de interação com os usuários. Certamente é preciso que a sintaxe da interface se assemelhe o máximo possível da sintaxe da LN. Mas, além disso, se as palavras formais têm um significado próximo ao seu significado natural então a comunicação se estabelece muito mais facilmente, porque não é exigido do usuário nenhum conhecimento semântico adicional. É importante também que a interface possa processar um conjunto significativo de sinônimos a fim de ganhar flexibilidade nas construções.

Um outro aspecto importante é que, numa interface inteligente, muitas vezes é possível reconhecer a intenção por trás dos enunciados, mesmo que estes sejam incompletos ou apresentem certos erros sintáticos. Neste sentido, sentenças curtas elípticas, isto é, incompletas, podem permitir que os usuários elaborem suas *queries* de forma incremental, como naturalmente fazemos em muitas situações. As sentenças com pequenos erros sintáticos podem ser reconhecidas se o restante da frase for

suficientemente não ambígua.

Finalmente, como já dissemos, em casos de ambiguidades que a interface definitivamente não consegue tratar, é sempre possível, diríamos mesmo que é uma atitude tipicamente humana, estabelecer um diálogo com o usuário a fim de resolver o impasse. Este procedimento é realmente adotado por muitas ILNs.

O processamento de informações armazenadas em bancos de dados pode se dar para efeitos de *consulta* e/ou *atualização* dos dados. Neste trabalho analisaremos uma interface voltada para a *consulta*, uma vez que o banco de dados criado a partir das pesquisas da Produção Agrícola Municipal (PAM) do IBGE têm um caráter permanente, estático, isto é, não constituem informações mutantes ou revisíveis, em virtude da própria natureza histórica dos dados.

O IBGE é um exemplo entre as muitas instituições e empresas que utilizam sistemas computacionais para disseminar informações, e que têm reclamado a necessidade de recursos eficientes para simplificar a interação usuário-máquina, recursos que possam promover a divulgação e a consulta dos dados para usuários inexperientes e diversos. Eis mais uma motivação para ampliar a área de ILN para consulta a bancos de dados.

Contudo, não podemos esquecer que a aplicação de LN para atualização de informações é também uma área de extremo interesse nas pesquisas, porque trata, entre outros aspectos, da forma declarativa das sentenças, que normalmente são consideradas estruturas básicas nos formalismos gramaticais mais conhecidos, tais como Redes de Transição e Gramáticas Lógicas, entre outros.

Arquitetura das Interfaces para Bancos de Dados

Em geral, as Interfaces em Linguagem Natural para acesso a banco de dados possuem internamente três fases de processamento:

- Análise Sintática - produz uma estrutura sintática para a sentença, gerada a partir da gramática adotada. Os analisadores sintáticos, ou *parsers*, são baseados em diversos formalismos gramaticais, dentre os quais podemos citar "Redes de Transição" [Woods - 1970, 1973], "Gramáticas Lógicas" [Pereira & Warren - 1980], "Gramáticas Semânticas" [Brown & Burton - 1975; Hendrix - 1978], e outros. Geralmente, a análise sintática é precedida de uma análise léxica, que consiste, basicamente, na consulta ao dicionário de vocábulos reconhecidos pelo sistema com o objetivo de obter informações de caráter gramatical e semântico.

- Interpretação Semântica - gera a forma lógica da sentença, isto é, uma representação intermediária entre a maneira pela qual o usuário pensa e a forma pela qual a informação pode de fato ser recuperada no banco de dados. Comumente, a forma lógica é semelhante às fórmulas do cálculo de predicados de primeira ordem (FOPC), acrescida de alguns operadores especiais para tratar aspectos sintático/semânticos da LN.

- Formalização - produz a representação final, geralmente na forma de linguagem de *query*, mapeando o resultado das análises anteriores em comandos de acesso ao banco de dados. Nesta etapa algumas funções pragmáticas costumam ser incluídas.

De uma forma geral, os sistemas para processamento de LN operam através de um *processo incremental*, isto é, o sistema passa o resultado parcial de um nível de análise ao nível seguinte. Por exemplo, antes de terminar toda a análise sintática de uma sentença, é possível passar informações parciais ao interpretador semântico, criando uma certa simultaneidade entre as análises sintática e semântica. Esta técnica traz alguns benefícios, dentre os quais a eliminação prematura de certos tipos de ambiguidades, conforme veremos mais adiante. A divisão em três fases de análise é mais uma distinção conceitual do que sequencial.

De fato, a maioria dos sistemas combina as duas primeiras fases - gramatical e semântica - em uma única fase, como também acontece neste sistema. Na interface AGRON as duas primeiras fases se confundem devido ao caráter semântico codificado nas regras da gramática.

No aspecto funcional, espera-se de uma interface em LN certas características que buscamos incorporar ao AGRON, tais como:

- aproximar-se ao máximo da sintaxe e da semântica da língua;
- não requerer do usuário conhecimento detalhado do banco de dados;
- interpretar perguntas incompletas ou com pequenos erros;
- reconhecer e resolver ambiguidades e inconsistências lógicas.

Quanto à filosofia interna de operação e à forma de acesso ao banco de dados, podemos classificar a maior parte das ILNs conhecidas em duas categorias:

- lógicas - utilizam uma estrutura baseada em lógica e interpretam o banco de dados como um conjunto de hipóteses, como é o caso dos sistemas CHAT-80 [Warren & Pereira - 1980] e TEAM [Grosz et al - 1985]. Neste caso, a interface funciona como um provador de teoremas do tipo PROLOG e o banco de dados não possui uma forma clássica (hieráquica, em rede, ou relacional).
- não-lógicas - podem, eventualmente, usar uma forma lógica intermediária semelhante ao Cálculo de Predicados de Primeira Ordem, porém na fase de formalização acabam por traduzir a sentença numa linguagem de *query* para acesso a um banco de dados convencional. Este é o caso, por exemplo, das interfaces PARNAX [Guida, Solmavico et al - 1983], DONAU [Solmavico et al - 1980], SAPHIR [Normier, Zarri et al - 1985], e também da interface AGRON.

Existem vantagens e desvantagens na utilização de um ou outro tipo de estrutura. As interfaces lógicas são mais indicadas quando a necessidade é checar a validade ou não de um enunciado, contra um conjunto de premissas que compõem o banco de dados, ou seja, quando a demanda é por uma informação de caráter *qualitativo*. Neste sentido, as sentenças mais comumente tratadas são assertivas e perguntas sim/não, embora outras construções linguísticas sejam possíveis.

Por outro lado, as interfaces não-lógicas se aplicam mais adequadamente aos casos onde ocorre a premência por um dado *quantitativo*, pontual ou não, isto é, obtido diretamente no banco de dados ou resultante de operações sobre os dados armazenados, tais como soma, média, interseção ou união. Neste caso, os tipos de sentenças processadas são, geralmente, comandos imperativos e perguntas. Este é o contexto de utilização da interface AGRON, projetada para atender a necessidade de obter informações quantitativas das pesquisas estatísticas agrícolas realizadas no IBGE.

Interação Usuário-Máquina

No acesso a bancos de dados, e na interação usuário-máquina de um modo geral, é comum o tratamento dos enunciados como eventos isolados, ou seja, a interpretação das sentenças é, supostamente, independente das sentenças anteriores. Neste modelo, a comunicação entre o usuário e o sistema é feita na forma de uma pergunta ou comando que gera um acesso ao banco de dados, onde este acesso não depende de comandos prévios.

Contudo, existe uma outro paradigma, mais próximo da realidade humana, que afirma que os enunciados não são entidades separadas. Em muitos situações ocorrem construções que exigem um cruzamento com enunciados anteriores, isto é, determinadas sentenças só adquirem significado no contexto. Neste caso, o comando seguinte traz algum tipo de referência ao comando anterior, ou então é uma uma complementação, uma reformulação do precedente, criando uma estrutura de diálogo, e estabelecendo uma dependência contextual entre as sentenças. Este tipo de referência é dita anafórica intersentencial.

Para atender as necessidades de processar enunciados interelacionados, algumas implementações de ILNs acrescentam facilidades para tratamento de referências pronominais e processamento de elipses, entre outras. Neste trabalho, veremos o caso particular do processamento de elipses segundo a proposta de Frederking para o sistema PSLI3 [Frederking - 1986], e avaliaremos o benefício da utilização desta técnica na interface AGRON.

Numa visão clássica, a interação usuário-máquina pode ser classificada em quatro tipos de sentenças: declarativas, perguntas sim/não, perguntas *qu* e comandos imperativos.

1. Declarativas

São sentenças utilizadas com o objetivo de fazer uma afirmação, ou declaração sobre uma entidade, objeto, ou situação. São utilizadas nas interfaces em LN para bancos de dados quando os dados são flutuantes, ou seja, quando é necessário atualizar informações a partir de assertivas.

Geralmente são estruturas básicas em qualquer língua, e por isso, são consideradas fundamentais em muitos formalismos da Línguistica Computacional. Contudo, a ênfase na interface AGRON é a consulta de informações. Logo, discutiremos a forma declarativa somente quando for conveniente, sem nos preocuparmos em analisá-la mais longamente.

Exemplo: "A Universidade no Brasil é carente de recursos."

2. Perguntas Sim/Não

São perguntas cuja resposta básica é sim ou não. De um modo geral, expressam a necessidade de saber se determinado fato é verdadeiro. A estrutura da pergunta é semelhante a afirmativa correspondente, a menos do ponto interrogação.

Exemplo: "Existe pesquisa em Linguagem Natural na COPPE?"

Na verdade, por trás deste tipo de pergunta podem existir outras intenções dos falantes e aspectos inusitados. Pesquisadores de LN têm estudado formas de fornecer respostas mais adequadas e cooperativas. Uma boa fonte para estudo do problema de geração de respostas cooperativas para perguntas sim/não é o trabalho de Kaplan [1983], onde algumas técnicas são apresentadas.

Uma intenção típica é o interesse numa informação *consequente*. Considere as duas sentenças, uma pergunta e uma resposta:

"Sueli foi a biblioteca?"
"Não, ela está na sala de aulas."

Na resposta fornecida, a construção "ela está na sala de aula" é dita consequente - se ela não foi a biblioteca então ela foi a outro lugar, à sala de aulas. Mais exemplos sobre respostas cooperativas serão dadas na descrição da cobertura sintática da interface AGRON.

3. Perguntas *qu*

São perguntas iniciadas pelos pronomes *que*, *quando*, *quanto*, *quem*, entre outros, destinadas a obter todo tipo de informação. Têm a forma da declaração correspondente, exceto pela parte questionada que é substituída por um pronome ou locução interrogativa.

Cada tipo de pergunta *qu* equivale a omissão de uma determinada parte da sentença declarativa, segundo a condição ou situação que se questiona. Considere a seguir uma sentença declarativa e a pergunta *que* correspondente.

"Maria estuda inglês." (declaração)
"O quê Maria estuda?" (pergunta *quê*)

Perguntas *qu* caracterizam o **movimento em linguística**. O termo *movimento* surgiu com a gramática transformacional de Chomsky, em 1965. Ele afirmava que a análise estrutural das sentenças, defendida pelos behavioristas, não captava a criatividade essencial da linguagem humana. Os enunciados não aparecem simplesmente na natureza, nem são meramente a junção de constituintes gramaticais, mas são resultado da capacidade mental do orador. Daí a necessidade de examinar o que está por trás da faculdade de criar e compreender sentenças.

Além do mais, muitos linguistas modernos concordam que a análise estrutural de uma sentença é superficial, pois é determinada pela *ordem* dos elementos na sentença. Com base neste tipo de argumento, Chomsky propôs uma teoria linguística que relacionava o que ele denominou os dois níveis estruturais de uma sentença, as estruturas *superficial* e *profunda*. Trata-se de criar estruturas que não são observadas diretamente dos dados, mas apelam para uma realidade mental, para a intuição do orador. Podemos dizer, resumidamente, sobre os dois níveis da sentença:

- Estrutura profunda - é a estrutura gerada pela gramática transformacional; muitas vezes corresponde a forma declarativa das sentenças, e nem sempre é igual a estrutura da sentença original;

- Estrutura superficial/externa - é a estrutura próxima ou equivalente a estrutura da sentença apresentada; em geral resulta de transformações e/ou *movimentos* sobre componentes da estrutura profunda.

No exemplo anterior ("O que Maria estuda?), a transformação ocorre no objeto direto *inglês*, que é substituído pelo pronome interrogativo *que* e transladado para o início da sentença, caracterizando assim a forma da pergunta *que* associada.

Outra situação que caracteriza o movimento em linguística são as formas passivas. A forma passiva é marcada por uma locução verbal com o verbo auxiliar *ser* seguido de um verbo no particípio passado que admite um objeto, neste caso introduzido pela preposição *por*. Assim, a forma passiva da sentença "João ama Maria" é "Maria é amada por João". Aqui o movimento se dá na inversão do sujeito pelo objeto, além da transformação do verbo da voz ativa para a passiva. A forma passiva, no entanto, não é interessante no contexto do sistema AGRON, porque não constitui uma construção comumente usada para consultas.

Para as gramáticas que são baseadas na forma declarativa das sentenças, o problema das perguntas *qu* é como lidar com o fato de que embora um componente da frase declarativa correspondente seja omitido a frase continua válida. Uma solução discutida por Allen [1987], adequada inicialmente para o formalismo de Rede de Transição Aumentada (ATN) [Woods - 1970], seria a de armazenar os termos interrogativos na forma de registros numa "lista de espera" e utilizá-los mais

tarde para preencher lacunas (*slots*) na análise dos constituintes gramaticais da sentença. Além disso, a gramática tem que ser *aumentada* com um mecanismo que verifique se o termo interrogativo da "lista de espera" pode ser substituído pelo componente que falta. Considere o exemplo a seguir:

"Mamãe foi ao supermercado ontem." - declaração
"Quando mamãe foi ao supermercado?" - pergunta "quando"
"Mamãe foi *quando* ao supermercado?" - seria analisada assim

Outros sistemas, como o AGRON, contornam o problema produzindo explicitamente uma subgramática para perguntas *qu*, ou seja, aumentam o número de regras para reconhecer sentenças que tenham a forma gramatical das perguntas *qu*. Na verdade, a interface AGRON, pela sua aplicação destinada a consulta em bancos de dados, possui uma gramática voltada para perguntas e comandos imperativos, não possuindo regras gramaticais para reconhecer, por exemplo, formas declarativas e passivas.

O aumento da gramática pode ser feito através de *metaregras* que, dado uma gramática simples, vão gerar uma nova gramática estendida. As metaregras funcionam, até certo ponto, como transformações, porém são limitadas a uma única regra de cada vez, enquanto as transformações podem operar em árvores sintáticas arbitrárias. Esta é uma técnica usada na teoria da Gramática de Estrutura de Frases Generalizadas (GPSG) [Gazdar et al - 1985].

4. Comandos Imperativos

São sentenças que caracterizam algum tipo de ação do sistema com o objetivo de fornecer a informação solicitada pelo usuário. Geralmente são iniciadas com um verbo imperativo, tais como *imprima, liste, forneça, identifique, selecione,* e outros.

Exemplo: "Liste a quantidade de artigos sobre Linguagem Natural na biblioteca do Núcleo de Computação Eletrônica."

Este tipo de sentença da LN é a forma mais próxima da estrutura dos comandos aceitos nas linguagens de computação para *query* a bancos de dados. Logo, os comandos imperativos são as sentenças que, em geral, podem mais facilmente ser mapeadas para a forma final das sentenças nas ILN para acesso a bancos de dados.

Além das perguntas sim/não, perguntas *qu*, e comandos imperativos, existem ainda outras formas de solicitar informação. É o caso das perguntas *como* e *por quê*, porém estas são, geralmente, difíceis de responder. A dificuldade básica reside no conhecimento que deve ser atribuído ao questionador, tema de pesquisa das Teorias e Modelos de Crença. Neste caso, significa que a resposta fornecida pode ser satisfatória para um usuário que detenha determinado conhecimento, mas pode ser insuficiente para outro usuário que não seja possuidor do mesmo conhecimento. A interface AGRON, na sua proposta inicial, não processa perguntas *como* e *por quê*.

III. TRABALHOS RELEVANTES

Esta seção se dedica a uma revisão sobre sistemas relevantes na área de Interfaces em Linguagem Natural para acesso a Bancos de Dados. Neste caso, supomos que o leitor já possui algum conhecimento dos conceitos, técnicas e formalismos utilizados nesta área. Caso contrário, recomendamos a leitura desta seção depois do restante do trabalho, após a qual supomos que haverá uma base de informações suficientes para compreensão da maior parte das técnicas aqui referenciadas.

LUNAR

A interface LUNAR [Woods et al - 1972] foi uma das primeiras interfaces desenvolvidas para acesso a bancos de dados. O LUNAR é utilizado na NASA para consultar um banco de dados com informações sobre amostras lunares trazidas pelo programa espacial Apollo.

Uma das características mais interessantes do LUNAR é a utilização de um método inventado por Woods [1968] conhecido como "semântica procedimental". O objetivo deste método é tornar executável a forma lógica, ou uma interpretação intermediária semelhante. Isto significa que esta representação intermediária pretende ser de fato interpretada como um programa.

Para tanto, o método baseia-se na interpretação dos quantificadores da forma lógica. A cada tipo de quantificador identificado correponde um procedimento específico que traduz o seu significado na forma diferenciada de acesso as informações do banco de dados. O resultado do procedimento é então utilizado para compor a resposta.

A estrutura básica do LUNAR é composta de um analisador sintático, que utiliza o formalismo de ATN (Rede de Transição Aumentada), cobrindo uma variedade razoável de perguntas, e de um interpretador semântico, que gera uma forma lógica executável usando o método da "semântica procedimental".

A maior parte das *queries* no LUNAR é caracterizada por comandos imperativos. Assim sendo, a forma lógica permite ações, do tipo *listar*, *selecionar*, e outras. Além dos quantificadores mais comuns tais como *todo*, *algum*, *cada* e outros, a forma lógica também admite certos operadores numéricos - *três*, *mais que quatro* - e alguns operadores aritméticos - *média*, *soma*.

CHAT-80

O sistema CHAT-80 [Warren & Pereira - 1980] é uma interface em LN baseada em lógica. Desta forma, o banco de dados é visto como um conjunto de hipóteses ou assertivas lógicas e a consulta se torna um processo de prova de teoremas.

Neste caso, a estrutura do banco de dados elimina algumas restrições que ocorrem nos modelos tradicionais (hierárquico, rede e relacional). Por exemplo, um banco de dados relacional normalmente não representa fatos gerais, tais como "todo vôo serve uma refeição", embora esta informação possa ser obtida consultando todas as instâncias de vôo. Mas uma declaração baseada numa fórmula lógica pode admitir quantificadores universais para expressar tais fatos.

O CHAT-80 usa a linguagem PROLOG em todas as fases da análise, que são três: análise sintática, análise semântica e determinação do escopo de quantificadores. O resultado final é uma forma lógica expressa em DCW (definite closed world), uma notação desenvolvida especialmente para o sistema e que pode facilmente ser interpretada por um sistema de programação lógica.

O método de prova de teoremas utilizado no PROLOG é baseado em cláusulas de Horn. Mas as cláusulas de Horn não são adequadas para expressar raciocínios sobre negação. Ao invés de provar que uma fórmula (~P) é verdadeira, o sistema tenta provar que P é verdadeira e falha. O fato do sistema falhar ao tentar provar que P é verdade não garante que (~P) é verdade. Este mecanismo é conhecido como "*hipótese do mundo fechado*", isto é, no caso do sistema falhar ao tentar provar P, supõe então que (~P) é verdadeira. Esta suposição equivale a assumir que tudo que não está explicitamente declarado é falso.

Para contornar a situação, a linguagem DCW cria alguns mecanismos que estendem o poder de expressibilidade das cláusulas Horn. Um deles é a introdução do operador *unless*, onde (*unless* P) é verdade somente se o provador de teoremas tenta recursivamente provar P e falha. Este operador só pode ser usado no lugar do operador de negação quando a fórmula P não contém variáveis.

Uma cláusula DCW é da forma:

(RESPOSTA x1 ... xn) <= C

onde a condição C é uma conjunção de fórmulas e a resposta é o conjunto finito das tuplas x1 ... xn que satisfazem C.

A análise sintática do CHAT-80 utiliza técnicas de gramáticas lógicas. A estrutura sintática é transformada numa forma lógica através de uma interpretação semântica baseada em regras que utilizam uma classificação (taxonomia) dos argumentos dos predicados para selecionar os complementos aceitáveis.

Finalmente, um algoritmo para determinação do escopo dos quantificadores é aplicado usando heurísticas simples que fazem distinção entre determinadores do tipo *cada, todo, qualquer,* entre outros.

O CHAT-80 pode ser transportado para um novo domínio linguístico e para novas estruturas de arquivo desde que estas últimas possam ser mapeadas para cláusulas de Horn. As aplicações do CHAT-80 utilizam o banco de dados PROLOG interno, logo, a adaptação para um outro banco de dados requer uma interface entre este último e o PROLOG. Além disso, a adaptação deve ser feita por alguém que conheça a arquitetura do sistema.

PARNAX

O sistema PARNAX [Guida, Somalvico et al - 1983] foi implementado experimentalmente num Siemens 7748 do centro de estudos CSELT (Torino, Itália). O PARNAX é uma interface em LN projetada para banco de dados ADABAS, e voltada para a compreensão do idioma italiano.

A característica principal do PARNAX é o analisador sintático/semântico que utiliza uma filosofia bastante aceita pelos psicolinguísticos em geral, que é o processamento simultâneo do conhecimento sintático e semântico.

O processamento semântico é feito em duas fases sequenciais: uma parte não determinística que valida e completa a atividade do analisador sintático, e uma parte determinística que gera uma representação interna intermediária chamada *metanatural*. Um algoritmo paralelo gerencia a cooperação entre o analisador sintático e a parte não-determinística do analisador semântico.

O analisador trabalha em dois níveis: uma macro-análise, que considera a estrutura superficial/externa da sentença e sugere uma fragmentação da mesma, ao mesmo tempo que compõe os fragmentos já analisados; e uma micro-análise, efetuada em três fases e cujo produto final é uma estrutura *metanatural* independente do esquema lógico do banco de dados.

Na micro-análise, a primeira fase compreende uma análise léxica não-determinística. A segunda fase consiste de dois processos paralelos: uma análise sintática (baseada em gramáticas de dependência) e a primeira parte da análise semântica. A terceira fase é um processo determinístico de interpretação semântica que opera através da ativação bottom-up de módulos semânticos referenciados pelas árvores de estruturas sintático-semântica, associando valores as variáveis semânticas correspondentes.

Em resumo, a cada passo da análise, isto é, para cada fragmento da sentença que é processado, o analisador sintático propõe um conjunto de estruturas sintáticas válidas, ao mesmo tempo que o analisador semântico não-determinístico sugere um conjunto de estruturas semânticas possíveis. Dentre estas estruturas semânticas produzidas, somente aquelas que podem ser associadas a uma estrutura sintática prosseguem nas fases posteriores. Desta forma, procura-se diminuir o espaço de busca que é de fato muito grande durante a análise.

Terminado o processo de análise, um módulo chamado "formalizador" transforma a representação *metanatural* num programa NATURAL que pode então ser submetido para acesso ao banco de dados ADABAS.

O PARNAX parece ter uma cobertura linguística razoável do italiano, e pode tratar sentenças mal-formadas, elipses, formas telegráficas, um conjunto de referências anafóricas e *queries* múltiplas. Em casos de interpretações ambíguas, o PARNAX estabelece um diálogo limitado com o usuário exibindo paráfrases das possíveis interpretações e solicitando a escolha do usuário.

A utilização de algoritmos paralelos que intercalam processos de análise sintática e semântica no PARNAX sugere um grau de eficiência aceitável para a configuração em questão (em média 3 segs. de CPU para uma sentença de 9 palavras, sem o acesso ao SGBD). Por outro lado, mais importante ainda pode ser o fato do uso de algoritmos paralelos marcar uma linha de pesquisa promissora no desenvolvimento de Interfaces em Linguagem Natural.

DONAU

O sistema DONAU (Domain Oriented NAtural language Understanding) [Somalvico et al - 1980] foi desenvolvido no laboratório de projetos de robótica da Politécnica de Milão (Itália) com a finalidade de programar o robô SUPERSIGMA para a manipulação de objetos. Mais tarde, uma segunda versão do DONAU foi desenvolvida utilizando um outro domínio semântico e voltada para acesso a bancos de dados através do italiano. O DONAU foi, então, implementado num banco de dados do IIASA (International Institute for Applied System Analysis) que continha informações sobre os diferentes tipos de fonte de energia do mundo.

A evolução da primeira versão do DONAU para a segunda foi possível porque o problema de recuperar informações em um banco de dados é bastante semelhante aquele de se programar um robô, no sentido que as diversas fases do processo de compreensão da Linguagem Natural são aplicáveis em ambos os casos.

Embora a readaptação do DONAU a um novo domínio semântico não pareça trivial, a estrutura modular do sistema permite definir que partes são independentes do domínio e que módulos devem ser alterados. Porém, a readaptação deve ser feita por alguém que conheça o sistema.

A análise de uma sentença segue uma ordem tradicional nas ILN. Primeiro, a sentença é analisada sintaticamente através do sistema PIAF, desenvolvido na universidade de Grenoble, França, e que trata também o italiano.

Em seguida, o algoritmo de análise semântica utiliza uma rede de discriminação semântica, que é considerada uma estrutura importante na arquitetura do sistema. A análise semântica é baseada num conjunto significativo de sentenças chamado "protocolo de interação". Todas as peculiaridades, ambiguidades e problemas relacionados à compreensão da LN são tratados através deste protocolo de interação. O sistema foi desenvolvido com o objetivo de permitir a construção de uma rede de discriminação diferente para um outro domínio semântico.

Um terceiro módulo extrai as informações ditas "operativas" através da combinação de resultados entre as possíveis estruturas sintáticas e a rede de discriminação semântica, procurando eliminar ambiguidades.

A fase seguinte, chamada de "controle de ilegalidades", processa a informação obtida anteriormente eliminando as "ilegalidades", isto é, as informações que estão corretas sintática e semânticamente, porém são inconsistentes com a configuração do banco de dados. Este módulo trabalha basicamente com a adequação ao domínio semântico escolhido, gerando uma representação executável da sentença.

Utiliza-se como linguagem de *query* ao banco de dados a linguagem MICROPLANNER (LISP). Naturalmente, se uma outra linguagem for utilizada é necessário fazer as devidas alterações no sistema para mapear a forma final da nova linguagem.

TEAM

A interface TEAM [Grosz et al - 1985] foi desenvolvida no centro de Inteligência Artificial SRI International, Califórnia (EUA). O projeto é voltado para o desenvolvimento de uma interface em LN para banco de dados efetivamente transportável, isto é, que possa ser rápida e facilmente adaptada a novos domínios ou novas estruturas de banco de dados por alguém que não seja especialista em processamento de LN. O sistema tem dois modos de operação: aquisição e processamento de sentenças em LN.

1. Aquisição

Neste modo são extraídas as informações sobre a estrutura do banco de dados, as palavras e frases que referenciam os objetos e as relações entre estes objetos. A aquisição é operada pelo especialista em banco de dados (DBE) e dirigida através de menus (providos de help) que permitem que o DBE especifique ou altere uma gama de características linguísticas associadas ao domínio em questão, características estas que possibilitam mais tarde a inferência das respostas adequadas no modo de processamento de sentenças.

O TEAM processa três tipos de campo:

- aritmético: valores numéricos;
- simbólico: valores alfanuméricos geralmente correspondendo a substantivos ou adjetivos;
- característica: valores booleanos (verdade ou falso).

O processo de aquisição envolve três componentes importantes do sistema: o léxico, o esquema conceitual e o esquema do banco de dados.

O Léxico

É um repositório de informações sobre cada palavra e possui dois tipos de ítens básicos: fechados e abertos. Os ítens fechados são pré-definidos e compreendem pronomes, conjunções e determinadores (artigos, preposições); têm uma função gramatical específica e possuem pelo menos um significado independente do domínio. Os ítens abertos podem ser adquiridos e compreendem substantivos, verbos e adjetivos cujos significados dependem do banco de dados em questão.

A cada entrada léxica são associados nomes de campo, possíveis valores, relações, além de sinônimos, e outras informações sintáticas e semânticas que conferem os significados da palavra.

O Esquema Conceitual

É uma espécie de rede semântica que descreve o domínio do banco de dados e compreende informações sobre os objetos, suas características e relações, armazenadas sob a forma de hierarquia de classes (ou tipos) além de descrições de predicados não classificáveis, tais como predicados associados a adjetivos comparativos (por exemplo, "ser alto").

A hierarquia de classes é uma rede que expressa a taxonomia dos objetos. O TEAM fornece uma hierarquia de classes pré-definida que pode ser aumentada ou modificada pelo DBE conforme a necessidade de adequação a um domínio específico. A cada classe são associados predicados que estabelecem relações entre as classes. Por exemplo, o predicado capital-de (*capital,país*) relaciona as classes *capital* e *país*. Classes que correspondem a campos aritméticos contendo medidas incluem informações sobre a unidade de medida (metros, anos, e outras) e o tipo de medida (linear, temporal). As descrições de predicados não classificáveis contém informações complementares tais como grau e direção, por exemplo '+' para "mais alto que".

O Esquema do Banco de Dados

Descreve o mapeamento entre os predicados que aparecem na representação lógica interna da sentença (isto é, na forma lógica) e as relações do banco de dados. Note-se aqui a suposição de que o banco de dados a ser mapeado tem uma estrutura relacional.

É também no processo de aquisição que são especificadas pelo DBE as relações ditas virtuais, definidas através de *join* de campos de relações existentes. Na verdade, qualquer sentença que se refira a uma relação (virtual ou explícita) que não foi prevista e definida anteriormente, não será processada.

2. Processamento de Sentenças em LN

O modo de processamento de sentenças em LN é operado indistintamente por qualquer tipo de usuário que deseje acessar as informações do banco de dados.

Este módulo subdivide-se em dois componentes: o sistema DIALOGIC [Groz et al, 1982] que mapeia as expressões em LN para representações lógicas formais; e o tradutor de esquemas que transforma a forma lógica em comandos para o banco de dados.

O sistema DIALOGIC compreende: o analisador DIAMOND [Paxton, 1974], a gramática DIAGRAM [Robinson, 1982], o léxico, funções de interpretação

semântica, funções básicas de pragmática e procedimentos para determinação do escopo de quantificadores. O DIALOGIC utiliza as informações do léxico e do esquema conceitual.

Embora uma convergência de pensamento de teóricos indique a preferência por uma análise sintática/semântica simultânea, a opção no TEAM foi por um processamento sequencial. O argumento, segundo os autores do TEAM, é que separando-se a análise sintática da tradução semântica é possível o desenvolvimento independente da cobertura gramatical e da habilidade semântica.

A forma lógica do TEAM é uma lógica de primeira ordem estendida com alguns operadores de ordem maior e alguns quantificadores especiais para locuções interrogativas e determinadores.

A forma lógica gerada pelo DIALOGIC é transformada pelo tradutor de esquemas em comandos de uma linguagem de *query* chamada SODA [Moore, 1979] que tem interface para diversos SGBDs. O tradutor de esquemas usa o esquema conceitual e o esquema do banco de dados para executar tal operação.

O tradutor de esquemas pode ser visto como um sistema que reescreve a forma lógica através de transformações sucessivas que preservam a equivalência lógica (dentro da "hipótese do mundo fechado"), gerando uma forma mais adequada que produzirá as mesmas respostas quando avaliada no banco de dados. Assim como no CHAT-80, as fórmulas intermediárias obtidas durante as transformações se assemelham à cláusulas PROLOG e de fato são processadas como tal.

Finalmente, os comandos são submetidos ao interpretador SODA que executa-os e exibe a resposta.

O TEAM foi implementado numa máquina LM-2 LISP e inclui um interpretador PROLOG próprio para a tradução de esquemas.

Embora considerada eficiente e possuindo o grande trunfo da transportabilidade, a interface TEAM apresenta algumas restrições dentre as quais vale ressaltar a hipótese de que o banco de dados *não* pode ser reestruturado.

SAPHIR

Além de ser um sistema transportável para acesso a banco de dados em LN, o SAPHIR+RESEDA [Normier, Zarri et al - 1985] pretende ser uma interface inteligente, isto é, acrescida de uma base de conhecimento e de uma máquina de inferência, permite, sempre que necessário, uma conversão automática da sentença de entrada numa outra que é "semanticamente próxima". Por "semanticamente próxima" entende-se que a resposta para a sentença transformada implica no que teria sido a resposta da pergunta original.

O sistema SAPHIR [Normier et al, 1984] foi desenvolvido em ERLI, na França, com o objetivo principal de ser uma interface para o francês transportável e independente do domínio. Posteriormente, idealizou-se a incorporação de uma base de conhecimento e de uma máquina de inferência baseadas no sistema RESEDA [Zarri, 1984] a fim de tornar a interface mais cooperativa.

A sentença é submetida a um analisador sintático do francês. Algumas vezes o analisador recorre a diálogos simples para esclarecer ambiguidades. Como resultado da análise, é produzida uma representação na forma de predicados.

O sistema dispõe de um modelo da estrutura do banco de dados na forma de descrição de cálculo de predicados, próxima ao esquema conceitual. Assim, é possível checar se a resposta pode ser obtida diretamente do banco de dados ou não.

Se a solicitação procede, isto é, se a resposta pode ser mapeada diretamente, o sistema transforma a representação de predicados numa fórmula em linguagem formal de *query*. O SAPHIR possui traduções para QBE, SQL, ADABAS e CLIO. Caso contrário, o sistema tenta transformar a sentença em algo que tenha significado para o banco de dados, e que seja semanticamente próxima a sentença original.

Por exemplo, suponha um banco de dados com informações de pessoas e uma sentença do tipo "Quem já esteve antes nos EUA?", onde esta informação não está explicitamente armazenada. Ao invés disso, no banco de dados existe informação sobre onde cada pessoa nasceu, e sobre a escolaridade de cada um. O sistema é capaz de transformar a pergunta porque sabe duas coisas: se uma pessoa nasceu em um lugar então ela esteve neste lugar ao menos uma vez; para obter o grau universitário uma pessoa tem que cursar uma universidade e estar fisicamente no local desta universidade. Logo, sem saber explicitamente quem já esteve nos EUA, é possível garantir que as pessoas nascidas nos EUA ou que possuem um título de graduação de uma universidade americana certamente já foram aos EUA pelo menos uma vez. Segue deste exemplo que o sistema manipula o conhecimento do senso comum.

A descrição da base de conhecimento e da máquina de inferência associadas ao sistema e que caracterizam a inteligência da interface não serão abordadas neste trabalho. Contudo, vale a pena acompanhar os resultados do projeto SAPHIR, que podem ser promissores.

IV. UM PANORAMA DA INTERFACE AGRON

O subconjunto da linguagem processado por uma interface para LN deve ter um poder expressivo substancial. A linguagem compreendida pela interface AGRON é um subconjunto da língua portuguesa voltado exclusivamente para consulta em bases de dados que relacionam conhecimentos espaço-temporais e certos atributos de uma série de objetos que se deseja descrever.

No caso específico da implementação ora em vigor da interface AGRON, os objetos são os produtos agrícolas brasileiros considerados na pesquisa PAM (Pesquisa Agrícola Municipal) do IBGE, escolhida como pesquisa modelo para compor o banco de dados exemplo. Os conhecimentos espaço-temporais dizem respeito as informações de nivel geográfico e ano de coleta dos dados da pesquisa, respectivamente. Os outros atributos são as principais variáveis analisadas que integram o conjunto de dados para a série histórica da pesquisa, tais como área colhida e valor da produção.

Devido ao caráter puramente informacional, a interface processa basicamente comandos e perguntas com o objetivo de obter informações do banco de dados, fornecendo como saída o comando correspondente na linguagem de *query* SQL. Avaliaremos oportunamente as restrições impostas pela tradução para uma linguagem como o SQL.

A arquitetura do sistema é convencional. Existem duas etapas distintas de processamento: uma análise sintático-semântica das sentenças em LN, e uma fase de formalização, neste caso, de tradução para a linguagem SQL. Além disso, na versão ampliada do sistema, propomos a tradução para uma forma lógica intermediária segundo o modelo conceitual do banco de dados relacional alvo.

O sistema é essencialmente de processamento de sentenças, não havendo geração de Linguagem Natural. Portanto, o diálogo com o usuário só ocorre em raras situações, para esclarecimento de enunciados efetivamente ambíguos. Por exemplo, quando se menciona "Rio de Janeiro" sem uma qualificação prévia, é preciso saber se a referência é à cidade ou ao estado brasileiro.

A cobertura sintática e semântica da interface AGRON será fornecida em detalhes nas seções seguintes.

A Pesquisa Agrícola Mensal (PAM)

A PAM (Produção Agrícola Municipal) surgiu em 1938, no Ministério da Agricultura, com o objetivo de obter informações estatísticas sobre a atividade agrícola no Brasil, ficando a cargo do IBGE a coleta dos dados. A partir de 1973, o IBGE passou a responsabilizar-se também pela apuração e divulgação dos resultados. A divulgação era feita, principalmente, através de publicações padronizadas e do atendimento a pedidos específicos que davam origem a tabulações especiais. Mais tarde, alguns dados agregados (obtidos através de agregações dos dados básicos coletados) passaram a ser disseminados através do sistema SIDRA (Sistema de Informações de Dados Agregados do IBGE).

A pesquisa é realizada anualmente e abrange todo o território nacional, tendo o município como unidade de investigação. Os dados são obtidos por estimativas resultantes de informações prestadas por produtores, técnicos e representantes de instituições que atuam no setor agrícola. A pesquisa cobre cerca de vinte e nove produtos de culturas temporárias e trinta e três de culturas permanentes. As informações relativas a produtos pesquisados que não atingem uma tonelada de quantidade produzida e/ou um hectare de área colhida não são registradas no questionário.

As principais variáveis para divulgação são:

- área plantada
- área colhida
- produção obtida (quantidade produzida)
- valor da produção

Com o passar do tempo, verificamos uma crescente demanda de dados que dão origem às tabulações especiais, ou seja, informações que não constam de publicações e/ou não são oferecidas diretamente pelos meios de divulgação convencionais. O atendimento nestes casos é dirigido a um usuário particular, o que muitas vezes implica na elaboração de programas ou procedimentos novos para processar as informações do acervo. Isto pode tornar a execução do pedido impropriamente lenta, o que desaponta o solicitante e também prejudica o IBGE, pois a insatisfação dos usuários em virtude da morosidade no atendimento pode contribuir para uma imagem negativa da instituição.

Assim sendo, é de extremo interesse o desenvolvimento de métodos e técnicas que proporcionem uma forma de divulgação mais rápida e eficiente da enorme quantidade de dados armazenados no acervo do IBGE.

A partir de 1991, desenvolvemos o BMA (Banco de Microdados Agropecuários) com a finalidade de centralizar, numa poderosa estrutura de banco de dados, todas as informações das pesquisas da produção agrícola e pecuária mantidas pelo IBGE. Dois fatores contribuíram para o surgimento do BMA: a necessidade urgente de uma estrutura de dados que compatibilizasse toda a série histórica das pesquisas agropecuárias, facilitando o atendimento a tabulações especiais, aliada a aquisição de tecnologia adequada para manutenção dos arquivos de dados, neste caso, a compra do Sistema Gerenciador de Bancos de Dados DB2 da IBM.

O Banco de Microdados Agropecuários (BMA)

O Banco de Microdados Agropecuários (BMA) foi projetado com o objetivo de agrupar, em um único modelo de dados, as informações obtidas a partir das pesquisas PAM (Produção Agrícola Municipal) e PPM (Produção Pecuária Municipal) desde 1975, homogeneizando as classificações das variáveis e padronizando o tipo de ítem de informação disponível, a fim de viabilizar a construções de séries históricas que apresentem a evolução temporal do dado.

O BMA foi desenvolvido como um banco de dados relacional, utilizando o Sistema Gerenciador de Banco de Dados DB2, comercializado pela IBM. O acesso às informações é possível em um ambiente IBM/MVS, através de aplicações *batch* ou *on-line* que utilizam, basicamente, a linguagem de *query* SQL (*Structured Query Language*).

Embora o BMA aglutine as informações da produção agrícola e pecuária, concentramo-nos somente no domínio semântico agrícola como modelo para desenvolvimento do protótipo da interface AGRON. Futuramente, espera-se que uma implementação plena do AGRON amplie o dicionário léxico da interface para englobar também a produção pecuária, transformando o sistema numa interface em LN para a produção agropecuária no Brasil.

Na verdade, como veremos mais adiante, uma proposta mais abrangente e avançada de extensão do sistema AGRON é a construção de um módulo para aquisição de semântica. Com isso, o sistema deixa de ser orientado a um domínio específico para ser uma "máscara" semântica, voltada para qualquer universo que se encaixe na estrutura de casos espaço-temporal da gramática.

Com a implantação do BMA solucionou-se parte do problema de aquisição e manipulação dos dados do acervo do IBGE, pois o banco oferece uma estrutura uniforme de recuperação eficiente num ambiente computacional adequado e atual. Porém, a manipulação dos dados exige um conhecimento razoável de programação na linguagem de *query* SQL. Desta forma, o BMA tende a ser utilizado somente por

determinados indivíduos, especialmente programadores, que conheçam SQL ou outras formas de interfaces específicas para o DB2. Muitos profissionais do IBGE envolvidos com as pesquisas agrícolas não são capazes de fazer uso diretamente do banco de dados para atender solicitações externas ou mesmo para consultas internas, uma vez que não dominam o conhecimento mínimo necessário de SQL. Assim sendo, persiste a necessidade de existir um sistema que possa ser utilizado por usuários inexperientes.

A fim de cercar o problema de forma mais objetiva e identificar as necessidades mais urgentes, procuramos os profissionais do Centro de Disseminação de Informações para que pudéssemos colher amostras do tipo de solicitação que chega ao IBGE, amostras estas que serviram de base para orientar as estruturas sintáticas que são processadas pela interface AGRON.

A seguir alguns exemplos de construções típicas que constituem modelos de sentenças compreendidas pela interface:

"Informe a área colhida de café para os municípios da Bahia onde a produção foi maior que 100 toneladas em 1980."

"Qual foi a produção de milho no Rio de Janeiro em 1990?"

"Quais são os estados produtores de uva?"

"Informe as regiões com área plantada de cacau maior que 100 hectares."

"Qual a área colhida de cebola em Minas Gerais?"

V. BASES DA LINGUÍSTICA COMPUTACIONAL

Do ponto de vista computacional, a gramática é uma especificação formal das estruturas permitidas na linguagem. Assim, a gramática descreve a forma sintática das sentenças e a variedade de construções linguísticas, isto é, responde a questões tais como: Como agrupar as palavras para formar uma sentença? Como estão relacionadas as palavras e os sintagmas de uma sentença? Que palavras modificam outras palavras? Quais as que têm importância central?

Certamente, a palavra é a unidade linguística básica. Mas, como classificar e/ou agrupar as palavras em constituintes gramaticais significativos? De um modo geral, os linguistas modernos identificam um conjunto básico de constituintes gramaticais numa sentença, dos quais consideraremos apenas os três principais: sintagmas nominais, verbais e preposicionais.

Sintagma Nominal (SN)

Os sintagmas nominais são usados para referenciar objetos, conceitos, lugares, eventos, e outras entidades. Compreendem pronomes, substantivos, numerais, adjetivos e quaisquer expressões que designem uma entidade. No caso de expressões, o SN é dividido em uma palavra principal, geralmente um substantivo, dito *cabeça*, e outras palavras que especificam e/ou qualificam a natureza do objeto denotado pelo substantivo.

Os especificadores, podem ser classificados em:

- quantificadores - indicam quantos objetos são referenciados. Compreendem a classe dos pronomes indefinidos. Exemplos: *cada, todo, qualquer, algum, nenhum, muitos, poucos*.
- demonstrativos - indicam a situação do objeto em relação ao falante (*este* ou *aquele*), se o objeto pode ser identificado unicamente no contexto (*um* ou *o*), ou se é preciso identificar o objeto para o ouvinte (*o qual, que*). Abrangem os pronomes demonstrativos, relativos, e os artigos.
- possessivos - indicam a possessão do objeto descrito. Compreendem os pronomes possessivos tais como *meu, seu, nosso*.
- numerais - indicam a ordem ou o número de objetos, e são ditos ordinais ou cardinais, respectivamente.

Geralmente um SN pode conter no máximo um especificador de cada tipo, isto é, um quantificador, um demonstrativo, um possessivo e um numeral.

Exemplos de sintagmas nominais na linguagem processada pelo AGRON são:

"a area plantada de feijão"
"os estados produtores de soja"
"os produtos da lavoura temporária"

Os qualificadores associados a um SN podem ser adjetivos ou outros sintagmas nominais usados como modificadores, ligados ou não ao substantivo cabeça através de uma preposição. Por exemplo, considere o seguinte SN:

"área colhida de café"

O substantivo cabeça é *café*, e a expressão "área colhida", que por sua vez também é um SN, funciona como modificador de café.

No sistema AGRON, assim como na maioria das ILNs, o tratamento dispensado ao uso de especificadores (quantificadores, demonstrativos, numerais e modificadores) constitui um aspecto importante da interface.

Sintagma Verbal (SV)

O sintagma verbal é a parte da sentença correspondente ao predicado, isto é, aquilo que se deseja declarar, perguntar ou descrever sobre uma entidade ou situação. Consiste de um verbo principal, também chamado de *cabeça*, e opcionalmente, é acrescido de verbos auxiliares e/ou um SN chamado de objeto do verbo.

Embora uma boa parte das sentenças, especialmente as sentenças declarativas, sejam tipicamente compostas de um SN e um SV, existem sentenças que possuem apenas o SV explícito. É o caso dos comandos imperativos. Um exemplo de SV num comando imperativo do AGRON é:

"Liste a área plantada de soja."

onde "liste" é o verbo, e "a área plantada de soja" é o complemento.

O processamento de verbos e de sintagmas verbais é fundamental no poder expressivo de uma interface. O processamento de verbos na interface AGRON será discutido e avaliado oportunamente.

Sintagma Preposicional (SP)

Geralmente, o sintagma preposicional é formado de uma preposição seguida por um SN, chamado objeto da preposição. O SP é usado para qualificar diferentes partes da sentença. Pode estar, digamos, qualificando o substantivo cabeça de um SN, ou sendo usado como complemento verbal. Exemplo:

"Liste os estados produtores de milho em 1980."

Nesta sentença ocorrem dois sintagmas preposicionais: "de milho" e "em 1980".

No caso da interface AGRON, os sintagmas preposicionais são, caracteristicamente, os casos espaço-temporais da gramática, correspondentes aos adjuntos adverbiais de lugar e de tempo, isto é, às especificações de nível geográfico e ano da pesquisa, que equivalem a responder as perguntas "aonde" e "quando" ocorre a informação procurada. Estas são as incidências de sintagmas preposicionais tratadas com maior ênfase no AGRON.

Contudo, podemos observar sintagmas preposicionais associados a denominação dos produtos, como é o caso de "chá da Índia", onde "da índia" é um SP que qualifica o chá. Aqui, porém, a relevância semântica da qualificação não é significativa a ponto de fazer com que se dê um tratamento especial ao sintagma preposicional. Neste caso, o SP é somente um complemento nominal que, inclusive, é tratado como uma unidade léxica juntamente com o substantivo cabeça que qualifica. Assim também acontece com as expressões "cana-de-açucar", "castanha-de-cajú", "côco-da-baía", "Fernando de Noronha", "unidade da federação", "valor da produção" e quaisquer outras que designem um objeto, ou um modificador de objeto, que têm uma identidade no domínio semântico e são, por isso, consideradas como uma única entrada léxica. Ao descrever o léxico, voltaremos a abordar a qualificação dos objetos no AGRON.

Gramática Livre de Contexto (GLC)

Em Linguística Computacional existe uma variedade de formalismos para descrever uma gramática. Um dos formalismos mais conhecidos é a Gramática Livre de Contexto (GLC), introduzido por Chomsky em 1956 e utilizado primeiramente na ciência da computação por Knuth, em 1968, na análise de linguagens de programação. Na verdade, Chomsky estabeleceu quatro tipos de gramática: tipo 0 - com o poder da máquina de Turing; tipo 1 - sensível ao contexto; tipo 2 - livre de contexto; tipo 3 - regular.

Nas Gramáticas Livres de Contexto são usadas regras recursivas, comumente chamadas de *regras de reescrita*, para descrever as estruturas aceitáveis e guiar a análise sintática. As regras de reescrita, também conhecidas como *produções*, são da forma $A \rightarrow B$, onde a seta ($\rightarrow$) pode ser lida como "leva em" ou "é substituível por"; A e B são sequências de terminais e/ou não terminais da gramática. As regras alternativas para um mesmo não-terminal podem ser indicadas pelo símbolo | que é lido como *ou*.

Os terminais são os símbolos que não podem mais ser decompostos, formando assim o conjunto de palavras ou símbolos da linguagem propriamente dita. Os não terminais são metasímbolos, isto é, símbolos usados nas regras de reescrita para representar certos construtos gramaticais. O não-terminal usado como símbolo inicial nas operações de reescrita geralmente é representado por S.

Numa operação de reescrita, a sequência do lado esquerdo A é substituída pela sequência do lado direito B. No caso das GLCs, o lado esquerdo da regra A só pode ter, no máximo, um não-terminal. Segue-se o exemplo de uma gramática para analisar frases simples com um sintagma nominal como sujeito, e um sintagma verbal com um verbo mais outro sintagma nominal como objeto direto.

1. S $\rightarrow$ SN SV
2. SV $\rightarrow$ verbo SN
3. SN $\rightarrow$ substantivo
4. SN $\rightarrow$ artigo substantivo

A seguir, uma sequência de derivações para a frase "João comeu o bife". A regra de reescrita usada em cada derivação aparece entre parênteses.

S $\rightarrow$ SN SV	(1)
S $\rightarrow$ substantivo SV	(3)
S $\rightarrow$ substantivo verbo SN	(2)
S $\rightarrow$ substantivo verbo artigo substantivo	(4)
S $\rightarrow$ João comeu o bife	

Rede de Transição Recursiva (RTR)

Outro formalismo gramatical muito difundido é o de Rede de Transição Recursiva (RTR), descrito por Woods [Woods - 1970, 1973]. Ambos os formalismos, RTR e GLC, são equivalentes em termos de capacidade de geração de sentenças, isto é, o conjunto de linguagens que cada um deles pode descrever é idêntico.

As Redes de Transição (sem recursão) equivalem aos autômatos finitos, e, de forma semelhante a estes últimos, descrevem a gramática através de um esquema de nós, representando os estados intermediários na formação de um sintagma, e arcos rotulados, que indicam palavras ou categorias de palavras que são permitidas na transição de um nó, ou estado, a outro. Uma sentença é aceita como válida se existe um caminho na rede que vai de um nó inicial (S), até um arco de saída (pop), onde a sequência de palavras na sentença obedece a ordem das categorias que aparecem nos arcos deste caminho.

A Rede de Transição Recursiva têm a característica adicional de permitir que um arco seja rotulado com uma referência que aponte para uma outra rede de transição, ao invés de categorias de palavras. Desta forma, admitindo a recursão, o formalismo se configura suficientemente poderoso para descrever as mesmas linguagens que as GLCs.

Gramáticas Aumentadas

Os métodos descritos acima na verdade se limitam a reconhecer uma sentença como válida ou não. Mas, em geral, o que se pretende numa ILN é obter uma análise estrutural mais completa da sentença, que estabeleça certas relações e dependências entre os constituintes, e que possa ser usada nas fases seguintes de análise. Esse objetivo pode ser alcançado aumentando ou estendendo os mecanismos através de testes, registros auxiliares, e outros recursos, dando origem assim às chamadas *gramáticas aumentadas*. A Rede de Transição Aumentada [Woods, Kaplan - 1973] e a Gramática Lógica Aumentada [Pereira, Warren - 1980], utilizadas, respectivamente, nos sistemas LUNAR e CHAT-80, são dois exemplos de gramáticas aumentadas.

Uma das técnicas de extensão de gramáticas consiste em desenvolver um mecanismo para armazenamento da estrutura sintática da sentença. No caso da interface AGRON, que utiliza uma gramática de casos semântica, o armazenamento da estrutura da sentença é crucial para a fase de formalização, pois é preciso associar os constituintes da sentença a uma estrutura de *casos* semânticos. A estrutura de *casos* da sentença pode ser representada através de um conjunto de *slots* (lacunas), ou variáveis, que posteriormente serão usadas para mapear o comando em SQL.

Outras características de extensão das gramáticas em geral, e também da gramática do AGRON, referem-se à forma como as informações léxicas são armazenadas, além dos testes e ações associadas às regras de reescrita. Os testes associados a uma regra podem examinar se a entrada corresponde aos constituintes do lado direito da regra, e as ações constroem a estrutura de *slots* adequada, segundo as restrições semânticas. Estes mecanismos serão vistos detalhadamente quando da descrição do léxico e do analisador gramátical do AGRON.

VI. COBERTURA SINTÁTICA DO AGRON

Considerando os problemas da disseminação de dados no IBGE que foram apresentados anteriormente, fizemos uma avaliação exaustiva das solicitações de usuários internos e externos à instituição a fim de discriminar os tipos mais comuns de consulta.

No caso da produção agrícola, verificamos que a demanda é caracterizada, principalmente, por questionamentos sobre informações quantitativas, ou seja, estimativas geralmente organizadas em tabelas obtidas a partir dos dados coletados nas pesquisas e armazenados no acervo na forma de um banco de dados, o BMA. Naturalmente, o fornecimento de informações estatísticas é um dos principais serviços do IBGE.

A partir da análise estrutural das sentenças típicas, definimos uma proposta de cobertura sintática da interface AGRON que abrange, na verdade, uma boa parte da variedade de construções linguísticas voltadas para um sistema de perguntas e comandos imperativos para acesso a um banco de dados relacional com informações sobre a produção agrícola ao longo dos anos nas regiões, estados, e municípios brasileiros.

Na análise das *queries* mais frequentes, começamos a delinear um esquema comum a maioria das questões: geralmente é necessário mapear uma função $f(o,e,t)$, onde o caracteriza um objeto de investigação, e e t são parâmetros que correspondem respectivamente às dimensões espaço e tempo. Daí surgiu a idéia de estruturar a gramática em casos semânticos específicos que analisaremos separadamente *a posteriori*.

Um dos aspectos mais enfatizados na cobertura sintática do AGRON é o processamento de enunciados elípticos, tido como uma das principais facilidades da interface.

Além disso, incluímos neste capítulo a descrição sobre a estrutura de *casos* da gramática, porque, embora relacionada ao aspecto semântico, esta estrutura define a forma geral das regras sintáticas.

Tipos de Sentenças

Dos quatro tipos básicos de sentença (declarativa, perguntas sim-não, perguntas *qu* e comandos imperativos) a interface AGRON processa duas formas - perguntas *qu* e comandos imperativos. Exemplos:

"Qual o valor da produção de uva nos estados da região Sul entre os anos de 1985 e 1989?"

"Selecione os produtos da lavoura temporária no Sudeste cuja produção foi superior a 10 toneladas."

As sentenças declarativas, no caso de interfaces para banco de dados, são úteis nos procedimentos de atualização dos dados, que não faz parte dos objetivos definidos para o sistema AGRON.

Por outro lado, as perguntas sim-não tem pouca aplicabilidade no atendimento de informações quantitativas. Ou seja, até que ponto é interessante processar uma pergunta do tipo "Existe produção de tomate no Paraná?" respondendo simplesmente sim ou não, sem fornecer, efetivamente, a quantidade produzida? Neste caso, a atitude mais conveniente é oferecer uma resposta mais cooperativa.

Para ilustrar o problema de respostas sim/não insuficientes ou incompletas, considere a pergunta:

"O vôo Miami-Rio do sábado chega `as 15:00 hs?"

Uma resposta *não*, simplesmente, é insuficiente, pois não fica claro o motivo, que pode um dos seguintes:

- o vôo não chega `as 15:00 hs;
- não existe vôo Miami-Rio no sábado;
- não existe vôo Miami-Rio;
- não existe vôo nenhum.

Além disso, em determinadas perguntas algumas suposições são feitas. Por exemplo, na pergunta "João gostou do filme?" a suposição é que "João viu o filme". Uma resposta *não* pode ser duvidosa, pois não é possível distinguir se o motivo foi "João não gostou do filme" ou "João não viu o filme".

Problemas deste tipo ocorrem quando uma simples resposta *não* é fornecida. Duas estratégias sugeridas por Allen [1987] para contornar estes problemas são:

- Apagar uma parte da pergunta na representação lógica interna e tentar processar novamente. Desta forma, se a nova *query* tiver sucesso, significa que não existia uma situação que satisfizesse a condição apagada na sentença original. No exemplo do vôo para Miami, equivale a tentar recuperar "Um vôo Miami-Rio no sábado", onde se omite a restrição "às 15:00 hs". Caso se obtenha sucesso nesta pesquisa, uma resposta, mais cooperativa, pode ser "Não, o vôo não chega às 15:00 hs, mas sim às 17:00hs".
- Substituir uma constante por uma variável quantificada existencialmente. Assim, eliminando-se a restrição da constante, pode-se fornecer uma resposta mais específica, que corresponda a uma outra instância da mesma variável. Ainda no exemplo do vôo para Miami, poder-se-ia pesquisar "Existe em algum dia da semana o vôo Miami-Rio?" Em caso afirmativo, a resposta para a sentença original poderia ser "Não, o vôo semanal para Miami não é no sábado, mas no domingo".

Classes Gramaticais

As classes gramaticais cobertas pela linguagem do AGRON são artigos definidos, numerais cardinais, pronomes indefinidos, pronomes relativos, pronomes interrogativos, preposições, conjunções coordenativas, substantivos, adjetivos e verbos. A tabela completa das entradas léxicas é fornecida nos apêndices B e C.

Na realidade, as classes gramaticais não tem utilidade na definição das entradas léxicas no AGRON, porque a gramática que utilizamos é uma gramática semântica, onde o papel semântico dos constituintes é o que importa. Veremos que a cada terminal da gramática está ligado uma *classe semântica*, que o associa às regras sintáticas. Contudo, mantivemos as classes gramaticais no léxico para facilitar uma eventual extensão que verifique as concordâncias, ou outro aspecto relacionado às classes gramaticais convencionais.

Na definição de substantivos haverá, ocasionalmente, informação sobre complementos previstos, tais como adjetivos, preposições ou outros substantivos qualificadores. Na prática, os adjetivos não serão tratados em separado, mas definidos juntamente com o substantivo que qualificam, formando uma única entrada léxica composta. Assim sendo, um mesmo constituinte pode estar definido com ou sem complementos de acordo com as diferentes formas que pode se apresentar. Por exemplo, existem três entradas léxicas que começam com o substantivo "arroz", a saber: "arroz", "arroz irrigado", "arroz sequeiro".

Verbos

A capacidade de compreender verbos é essencial numa boa interface em LN. Geralmente, as interfaces são capazes de reconhecer os verbos *ter* e *ser*. Por exemplo, considere a pergunta:

"Que estados tem a produção de café maior que 10 toneladas ?"

Contudo, outra maneira seria perguntar:
"Que estados produzem mais que 10 toneladas de café ?"

Logo, uma interface em LN deve ter uma cobertura verbal suficiente para permitir uma flexibilidade razoável das sentenças.

A compreensão de verbos envolve certos aspectos: a identificação de quantos argumentos o verbo possui e se são opcionais ou não; como estes argumentos são mapeados em casos verbais (sujeito, objeto, complementos, e outros); os tipos de sintagmas preposicionais que podem ser usados com o verbo; se as formas passiva, não acusativa e dativa são possíveis. Além disso, se o verbo possui formas irregulares, estas devem ser adicionadas ao léxico. Tratando-se de interfaces em LN para acesso a banco de dados, devemos ter também algum mecanismo que relacione o predicado verbal e seus argumentos às estruturas do banco de dados. Na interface AGRON, isto é feito quando traduzimos os *casos* semânticos nos atributos e condições sobre as tabelas do banco de dados relacional, e posteriormente nos segmentos de comando da linguagem de *query* SQL.

Os verbos podem ser categorizados segundo sua regência. Assim, temos que os verbos intransitivos não admitem sintagmas nominais como complementos; verbos transitivos diretos admitem um sintagma nominal, dito objeto direto do verbo, como complemento; verbos transitivos indiretos admitem um objeto indireto, geralmente um sintagma preposicional que indica o recebedor de alguma ação de transferência ou posse; e verbos bitransitivos, que admitem dois sintagmas complementares, o objeto direto e o objeto indireto.

Porém, a estrutura de complementos verbais também inclui outras cláusulas que podem seguir o verbo, tais como advérbios e sintagmas preposicionais diversos indicadores de local da ação, época do evento, instrumento da ação, e demais aspectos.

No âmbito da interface AGRON, os verbos são caracteristicamente transitivos diretos, além de permitirem complementos de local e época. Isto se deve a aplicação principal do sistema que é obter dados sobre um objeto num contexto espaço-temporal. No caso específico do domínio semântico da produção agrícola, o que se

pretende é adquirir informações sobre os produtos agrícolas das pesquisas PAM em determinados anos para certas regiões, estados ou municípios brasileiros.

Existem três situações verbais típicas processadas pela interface AGRON:

1. A sentença é um comando imperativo iniciado por um verbo no imperativo ou no infinitivo, tais como *apresente*, *informe*, *selecione*, *forneça*, *liste*, *mostre*, ou *apresentar*, *informar*, *selecionar*, *fornecer*, *listar*, *mostrar*, e outros.
   Exemplo:

   "Listar os valores da produção de algodão na Bahia entre os anos de 1980 e 1985."

   Neste exemplo, o verbo *listar* tem três complementos: o objeto direto "os valores da produção de algodão", o complemento indicador de local "na Bahia", e o complemento temporal "entre os anos de 1980 e 1985".

   O primeiro complemento, o objeto direto, precisa ser especificado obrigatoriamente, a menos que trate-se de um enunciado elíptico. No enfoque da gramática de casos, veremos que o objeto direto preenche o caso semântico *objeto*.

   Os dois últimos complementos, de local e época, são opcionais e corresponderão aos casos semânticos *espaço* e *tempo*, respectivamente.

2. A consulta é uma pergunta *qu* iniciada por um pronome interrogativo - *qual*, *quanto*, *quando*, *quem* - seguida de um verbo auxiliar *ser* ou *ter*
   Exemplo:

   "Quanto foi a produção de cana-de-açucar em Alagoas em 89?"

   Neste contexto, o verbo é apenas um elemento de ligação, e muitas vezes a sentença pode ser substituída por uma outra sem o verbo. No exemplo anterior a sentença sem verbo poderia ser:

   "Qual a produção de cana-de-açucar em Alagoas em 89?"

3. A sentença pode ser um comando imperativo ou uma pergunta, como nas situações anteriores, porém possui um verbo específico do domínio semântico, introduzido ou não numa cláusula relativa.

   Exemplos:

"Quais foram os estados que plantaram soja em 87?"
"Que estados plantaram soja em 87?"

Aqui o verbo *plantar* é um verbo ligado ao domínio da produção agrícola. Nestes casos, o tratamento é sustentado por uma regra semântica que relaciona a ação representada pelo verbo a um atributo do objeto de pesquisa. A regra semântica associada ao verbo *plantar* diz que se um produto tem a área plantada maior que zero para um determinado estado então este estado *planta* o tal produto. Esta regra também se aplica para os verbos *cultivar* e *semear*. Já no caso do verbo *produzir*, a regra semântica diz que a produção do produto precisa ser maior que zero, significando então que um estado *produz* o produto.

A aquisição de semântica para os verbos é um dos maiores obstáculos à independência da interface em relação ao domínio. Por isso, avaliaremos uma forma de aquisição de semântica como premissa fundamental para transformar a interface dependente do domínio numa interface relativamente transportável.

Partindo do princípio de que a interface AGRON é voltada para domínios históricos, isto é, com informações produzidas num intervalo de tempo do passado, deduzimos que, excetuando-se os verbos para comandos imperativos, os tempos verbais mais comuns são tempos pretéritos. Além disso, como a finalidade é processar comandos ou perguntas sobre um objeto particular dirigidas ao sistema, os verbos se apresentam sempre na terceira pessoa, do singular ou do plural.

Assim sendo, a cobertura verbal do AGRON abrange verbos no pretérito perfeito e imperfeito do indicativo, e também verbos no presente do indicativo, supondo-se, neste caso, que a ação se refere ao ano corrente. Além disso, são processados os verbos no imperativo que caracterizam os comandos ao sistema, conforme descrito acima na primeira situação verbal.

Para os verbos ligados ao domínio semântico da aplicação, os argumentos dos predicados verbais são valores ou operações sobre valores de algum campo em uma relação do banco de dados. Na verdade, a cobertura verbal do AGRON é dirigida à semântica espaço-temporal que caracteriza a estrutura de casos. Por isso, os verbos serão novamente abordados à luz dos casos semânticos.

Estrutura de Casos

As sentenças na linguagem do AGRON consistem de quatro segmentos ou sintagmas fundamentais:

- uma expressão definindo o objeto de pesquisa, que no banco de dados em questão é caracteristicamente um produto agrícola; mas o objeto também pode ser uma referência a região, estado, município ou ano;
- uma especificação opcional de época, neste caso, o(s) ano(s) para a consulta; em caso de omissão assume-se a consulta para todos os anos armazenados no banco de dados;
- uma expressão opcional indicadora do local, que pode ser um conjunto de regiões, estados ou municípios brasileiros, ou ainda uma expressão do tipo "os municípios do estado do Rio de Janeiro";
- uma cláusula relativa restritiva opcional, que age como um filtro atuando sobre os objetos de pesquisa a fim de limitar o universo de busca conforme desejado.

Além disso, a sentença pode ser iniciada por um indicador opcional de comando imperativo ou pergunta, geralmente na forma de um verbo imperativo ou pronome interrogativo, respectivamente.

Considere o exemplo a seguir:

"Informe a área colhida de café para os municípios do estado do Rio onde a quantidade produzida é maior que 100 toneladas."

Neste exemplo, "informe" é um indicador de comando imperativo, "a área colhida de café" é o objeto de pesquisa, "para os municípios do estado do Rio" é o indicador de local, e "onde a quantidade produzida é maior que 100 toneladas" é um filtro ou restrição para a busca. Não houve especificação de época, que poderia ter sido algo como "no ano de 1980".

Este tipo de construção caracteriza uma estrutura de casos *o quê*, *aonde*, *quando*, ou mais propriamente, *objeto, espaço, tempo*, fundamentada na teoria das gramáticas de casos, que será abordada mais adiante. Além desses, temos também um caso de *restrição*, correspondente a cláusula relativa restritiva, definido em função do problema de limitar o universo de busca no banco de dados. Os casos *espaço*, *tempo*, e *restrição* são opcionais, porém o *objeto* precisa ser sempre especificado, salvo quando a sentença é elíptica.

Caso Espacial - Hierarquia de Tipos

O *espaço* é caracterizado pela especificação de uma unidade ou um conjunto de unidades territoriais brasileiras, agrupadas em três níveis principais: nível municipal, nível estadual (ou de unidade da federação), e nível regional. O IBGE possui um sistema que gerencia as diversas divisões territoriais do espaço brasileiro, bem como algumas características de codificação e nomenclatura, sistema este que serviu de base para os códigos e nomes de município, estado e região utilizados na interface AGRON.

Uma das facilidades mais interessantes da interface AGRON é a possibilidade de processar uma hierarquia de *tipos* sobre o espaço territorial. Os níveis territoriais - município, estado, região - correspondem a tipos que definem relações entre si. Para maiores referências sobre hierarquia de tipos veja, mais adiante, a seção sobre representação do conhecimento.

Seja R o conjunto de regiões brasileiras, E o conjunto de estados, e M o conjunto de municípios. Existe uma hierarquia entre R, E e M nesta ordem, onde R é considerado mais abrangente ou superior a E, e, sucessivamente, E é superior a M. Para cada $r \in R$, existe um subconjunto $E_i \subset E$ tal que r e E_i são coincidentes, em outras palavras, um subconjunto de estados define uma região. Para todo e_i, se $e_i \in E_i$ então $e_i \underline{\in} r$, ou seja, se o estado está no subconjunto de estados que define a região, então dizemos que o estado *pertence* a região. Na verdade, esta relação não é uma relação de pertinência convencional da teoria de conjuntos, uma vez que esta última só ocorre entre um elemento e um conjunto. Por isso usamos a notação $\underline{\in}$ para distinguí-la da tradicional, mas mantemos o significado de pertinência porque, se por um lado r é um elemento de R no nível regional, r é um conjunto de estados no nível estadual.

O mesmo ocorre entre E e M: para cada $e \in E$, existe um subconjunto $M_i \subset M$ tal que e e M_i são coincidentes, e para todo m_i, se $m_i \in M_i$, então $m_i \underline{\in} e$. Ou seja, um subconjunto de municípios define um estado, e cada município deste subconjunto é dito pertencente ao estado.

Além disso, cada subconjunto de estados que define uma região são mutuamente exclusivos entre si, bem como os subconjuntos de municípios que definem estados são mutuamente exclusivos entre si. Isto é, para todo $i \neq j$, $E_i \cap E_j = \varnothing$ e $M_i \cap M_j = \varnothing$. O mesmo não acontece com outros níveis territoriais, tais como meso e microregião, que possuem interseções, porém estes não serão tratados na interface AGRON.

É importante notar que as relações entre os tipos preservam a transitividade, ou seja, se $m_i \subseteq e_i$ e $e_i \subseteq r$ então $m_i \subseteq r$. Assim, podemos falar, por exemplo, em "municípios da região Norte".

De uma maneira geral, a interface AGRON trata as seguintes relações:

- subconjunto de regiões;
- subconjunto de estados;
- subconjunto de municípios;
- subconjunto de estados de uma região;
- subconjunto de municípios de uma região;
- subconjunto de municípios de um estado.

Exemplos de expressões válidas:

"regiões do Brasil"
"regiões Norte e Nordeste"
"Sul"
"estados da região Sudeste"
"ufs da região Centroeste"
"unidades da federação do Brasil"
"estados do Rio de Janeiro e São Paulo"
"Minas Gerais, Espírito Santo e Bahia"
"municípios da região Norte"
"municípios do Sudeste"
"municípios do estado de Santa Catarina"
"municípios de Minas Gerais"
"municípios de Belém e Manaus"
"Salvador, Porto Seguro e Alcobaça"

Caso Temporal

O caso *tempo* se refere a um ano, ou conjunto de anos que identifica a época de interesse para consulta. Na verdade, a referência temporal poderia ser restrita apenas aos anos passados, quando houve coleta da pesquisa PAM (Pesquisa Agrícola Municipal) e a carga no banco de dados BMA (Banco de Microdados Agropecuários). Porém esta restrição limitaria a flexibilidade de certas construções, tal como o uso do tempo presente nas construções verbais.

Atualmente, o BMA possui informações para os anos de 1975 a 1991, logo, as consultas só têm sucesso se efetuadas dentro deste intervalo de anos. Contudo, para efeitos de implementação da interface AGRON, estes limites são irrelevantes. Na pior hipótese, o comando processado não devolverá os dados solicitados por falta de informações armazenadas. Assim, procedemos ao tratamento de verbos no presente, mesmo sabendo que não existem dados a partir de 1991.

O sistema AGRON reconhece o caso temporal introduzido por uma série de preposições ou expressões prepositivas. A seguir, exemplos de algumas construções válidas.

"no ano de 1977"
"para 85 e 86"
"entre os anos de 1979 e 1983"
"em 78"
"depois de 1985"

Assim como ocorre na hierarquia do caso espacial, também é possível uma extensão da semântica da interface para o processamento de subdivisões temporais, ou seja, meses, bimestres, trimestres, dias de um mês, e outras, conforme a necessidade da aplicação. A extensão depende da inclusão de novas regras gramaticais e classes semânticas para reconhecer construções temporais mais complexas, além da alteração no dicionário de lexemas para incluir as entradas associadas.

Caso Objeto - Hierarquia de Subsunções

O caso *objeto*, que constitui a entidade principal na pesquisa, possui um conceito que evoluiu com o tempo. Tradicionalmente, no IBGE dizíamos que os ítens de pesquisa eram compostos de uma parte referente a especificação de uma *variável quantitativa* e uma parte indicadora da *classificação* da variável. No sintagma "a área colhida de café", a expressão "área colhida" corresponde a uma variável quantitativa, e "café" é uma categoria da classificação dos produtos da lavoura permanente. Uma categoria pode estar associada a mais de uma classificação. Por exemplo, a categoria "café" também poderia estar associada a classificação dos produtos para exportação.

Em 1991, houve um estudo realizado por uma equipe de pesquisadores do IBGE e do *Statistics Canada* com o objetivo de automatizar a indexação dos ítens do acervo de dados. Uma das consequências deste trabalho foi justamente a introdução do conceito de *objeto* de pesquisa, generalizando a estrutura classificação-categorias. Atualmente os objetos estão organizados numa hierarquia de quatro níveis: assunto, grupo homogêneo, objeto composto e objeto simples. Para o objeto "área colhida de café", a hierarquia seria: agricultura > produção vegetal > produtos da lavoura permanente > café; e a expressão "área colhida" é, neste caso, a variável quantitativa associada a "café".

Além disso, adotamos a terminologia de *modificador* substituindo o antigo conceito de *variável quantitativa*. Assim, na interface AGRON, para formar o caso *objeto* temos a especificação de um objeto, simples ou composto, e um ou mais modificadores associados. Por exemplo, no sintagma "a área plantada e a área colhida de arroz sequeiro", as expressões "área plantada" e "área colhida" são modificadores do objeto simples "arroz sequeiro". Já no sintagma "o valor da produção dos produtos da lavoura temporária" temos "o valor da produção" como um modificador associado ao objeto composto "lavoura temporária".

Do ponto de vista da linguística computacional, os *objetos* do acervo de dados do IBGE formam uma estrutura de conceitos que vai além de uma simples hierarquia de tipos, porque necessitam de certas relações de qualificação e outras possibilidades semânticas para serem descritos. A hierarquia de objetos pode ser entendida como uma taxonomia de generalização/especificação, nos moldes do recente formalismo semântico de subsunções definido por Woods [Woods, 1991] a partir do sistema de representação de conhecimento KL-ONE [Woods & Brachman, 1978; Woods & Schmolze, 1985].

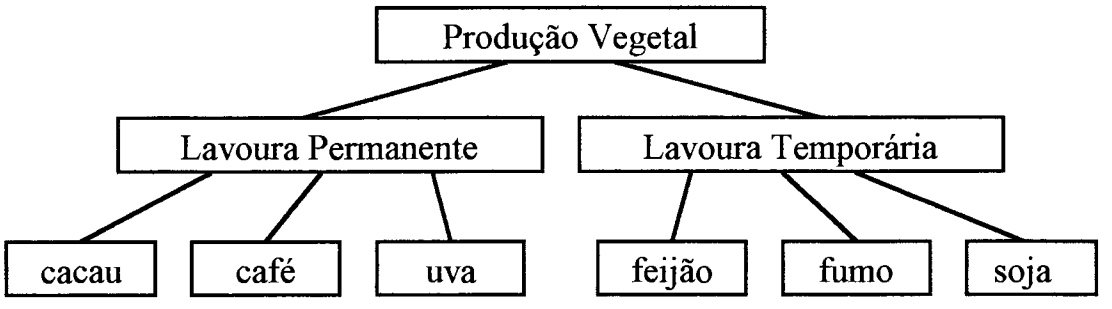
A característica mais importante do KL-ONE é a chamada generalização terminológica (*terminological subsumption*) onde um conceito estruturado é automaticamente classificado com relação à taxonomia de outros conceitos. Em linha

gerais, a taxonomia de conceitos é tal que os conceitos mais específicos se relacionam a conceitos mais gerais, ou seja, conceitos mais amplos *generalizam* conceitos mais específicos, e conceitos mais específicos *herdam* informações dos mais gerais.

Woods fornece uma definição de *conceito* independente da estrutura de dados, buscando captar o aspecto intensional da idéia, e para isso recorre a chamada *descrição abstrata*. A *descrição* é uma entidade conceitual abstrata que codifica certas características de coisas que podem existir ou não, isto é, de objetos que denotam alguma coisa "real" ou não. As descrições se dividem em dois tipos, atômicas e compostas, onde as compostas são definidas em função das atômicas. Partindo dessas premissas, Woods adota um esquema para representar as descrições ou conceitos, e se preocupa em discutir a semântica dos diversos *links* ou elos, distinguindo entre os elos assertivos e os estruturais, além de definir os operadores de quantificação, de formação de relações, e outros que ampliam o poder representativo do formalismo.

No caso da rede semântica de produtos agrícolas, as folhas da hierarquia correspondem a conceitos atômicos, indivisíveis, que não generalizam nenhum outro conceito, e os níveis superiores consistem em generalizações sucessivas a partir dos conceitos atômicos. É possível definir os elos estruturais e assertivos para uma tal rede semântica, além da aplicabilidade dos operadores utilizados na taxonomia de subsunções. Mas isto é trabalho para uma outra dissertação.

Para ilustrar o uso da hierarquia de subsunções, considere o seguinte exemplo. O conjunto de produtos da lavoura permanente é um objeto composto que equivale a um conceito ou descrição composta na taxonomia de Woods, e as categorias "cacau", "café", "uva", e outras que integram este conjunto de produtos, são ditas objetos simples, ou conceitos atômicos. Assim, "lavoura permanente" *generaliza* "banana", "café", "uva", assim como "produção vegetal" *generaliza* "lavoura permanente" e "lavoura temporária". Os elos neste caso são estruturais, ou definicionais, refletindo a estrutura de subsunções dos produtos agrícolas.



Por hora, a interface AGRON trabalha somente com os dois últimos níveis da hierarquia, objetos simples e compostos, armazenando no léxico a informação semântica de generalização/especificação dos objetos. O acoplamento de uma rede semântica de subsunções para processar todos os níveis da hierarquia e as construções semânticas decorrentes constitui uma das extensões propostas para o sistema.

Caso Restritivo - Cláusulas Relativas

A especificação opcional de uma *restrição* ou filtro é uma forma, muitas vezes necessária ou simplesmente conveniente, de limitar o universo de busca. A *restrição* é formada pela identificação, completa ou não, de um objeto, mais a especificação de uma relação e um valor, ou conjunto de valores, para comparação.

Seja o seguinte sintagma:

"cuja quantidade produzida é maior que 100 toneladas"

Neste exemplo, a identificação incompleta do objeto é "quantidade produzida", e a expressão filtro é "maior que 100 toneladas". Note que não foi mencionado o objeto propriamente dito, apenas o seu modificador, caracterizando uma situação de elipse. Neste caso, assume-se a instância do caso *objeto* corrente.

Assim sendo, considere agora o exemplo:

"Informe a área colhida de café nos estados onde a quantidade produzida é maior que 100 toneladas"

Aqui, o objeto subentendido da restrição é "quantidade produzida de café".

O que ocorre sintaticamente é que a interface AGRON é capaz de processar algumas frases relativas restritivas com alguns constituintes elípticos que, neste caso, são associados ao objeto da consulta referido anteriormente. As cláusulas relativas são identificadas por pronomes relativos tais como *que*, *cujo*, *cuja*, *onde*, entre outros.

Além disso, a expressão de comparação aceita operadores relacionais tais como *igual*, *maior*, *maior ou igual*, *menor*, *menor ou igual*, *diferente* e sinônimos.

Exemplos:

"Qual é o valor da produção de arroz para os municípios do estado de S. Paulo que tiveram área plantada superior a 30 hectares?"

"Informe a área plantada de soja nos municípios onde a produção foi maior que 100 toneladas."

Ordem de Casos - Movimentos Não-Locais

Na linguagem do AGRON, a ordem de especificação dos casos semânticos não importa. A inversão de casos pode ser verificada nos seguintes sintagmas:

"o valor da produção de arroz no ano 1984 em Goiás"
"o valor da produção de arroz em Goiás no ano de 1984"

Ambos são reconhecidos pelo analisador gramatical da interface AGRON.

Uma facilidade importante do algoritmo de análise é justamente não obrigar uma ordem na especificação dos casos gramaticais. Ou seja, o tratamento de movimentos não-locais ou arbitrários (*umbounded movements*) dos casos está implicíto na análise gramatical das sentenças.

Porém, certos movimentos não são permitidos como, por exemplo, a inversão dos constituintes dentro de um mesmo caso. Sejam as construções:

"Liste a área plantada e a área colhida de milho"
"Liste, para o milho, a área plantada e a área colhida"

Embora ambas tenham o mesmo valor semântico, a segunda não é reconhecida pela linguagem do AGRON, que possui, como restrição sintática, a precedência do modificador em relação ao objeto de pesquisa.

Ortografia e Concordâncias

Em geral, não haverão cheques de concordância nem verificações ortográficas, somente em ocasiões especiais para resolver certas ambiguidades. Numa ILN para acesso a banco de dados deste tipo a ênfase não é a correção gramatical, pois o objetivo maior é buscar identificar a *query* e a intenção do usuário, que importam mais do que a forma sintática apresentada.

Na implementação do protótipo da interface AGRON não existe a identificação de acentos ortográficos, nem a distinção entre maiúsculas e minúsculas. Na verdade, todas as palavras são traduzidas para letras minúsculas antes da consulta ao léxico. Isto acarreta algumas ambiguidades que normalmente não ocorreriam. Por exemplo, é necessário algum cheque ortográfico na palavra *para* para saber se denota a preposição ou o estado brasileiro, uma vez que, em virtude da limitação ortográfica, ambas mapeam a mesma sequência de caracteres. Neste caso, a estratégia utilizada para desambiguigação é a procura de letras maiúsculas: quando a ortografia indicar uma letra maiúscula no início da palavra assume-se o estado brasileiro *Pará*, caso contrário, entende-se uma referência à preposição *para*. Outro exemplo é a ortografia da palavra *são*, que precisa ser verificada para saber se a referência é ao verbo *ser* ou ao primeiro vocábulo do sintagma nominal *São Paulo*.

A ausência do tratamento de concordâncias permite sentenças mal-formadas sintaticamente tais como:

"Quais é o produtos da lavoura permanente em Sergipe?"

Apesar da falta de estilo e elegância linguísticas, do ponto de vista semântico a ausência de tratamento para as concordâncias não acarreta problemas de ambiguidade no domínio agrícola processado pela interface AGRON. Não encontramos nenhum enunciado onde o cheque de concordâncias fosse relevante para compreensão do sentido da frase, como no caso citado por Allen [1987] das sentenças:

"*Flying planes are dangerous*"
"*Flying planes is dangerous*".

Processamento de Elipses

O tratamento de enunciados elípticos é um dos principais aspectos na interpretação de LN na interface AGRON. A ênfase no processamento de elipses se deu em virtude da simplificação decorrente do uso de sentenças elípticas para solicitar uma informação, ou seja, o usuário não precisa reescrever certos constituintes da sentença quando estes se repetem na consulta seguinte.

Exemplos:
1: "Qual foi a área colhida, a área plantada e a produção de banana nos estados da região Nordeste entre os anos de 1970 e 1975?"
2: "E de laranja?"

No enunciado 2 foram omitidos praticamente todos os constituintes da sentença, exceto o objeto de pesquisa, que na sentença 1 era "banana" e na sentença 2 passa a ser um novo produto agrícola, neste caso, "laranja".

A economia de palavras nas sentenças elípticas é um dos fatores mais atraentes nas interfaces em LN quando comparadas às linguagens de *query* alfanuméricas. Nestas últimas é sempre necessário escrever todo o comando, porque a hipótese é de que os comandos são independentes uns dos outros.

Nos diálogos humanos, a elipse ocorre quando um orador omite uma parte do enunciado, ocasionando uma sentença incompleta, mal-formada localmente do ponto de vista sintático e semântico. Porém, o ouvinte deve ser capaz de inferir a porção omitida a partir de um conhecimento geral ou do contexto conversacional precedente [Frederking, 1986].

Na análise de sentenças elípticas em interfaces para LN, a hipótese subjacente é que o enunciado elíptico ou corresponde a um constituinte num enunciado anterior, ou introduz um novo constituinte que complementa algum enunciado prévio. Assim, no exemplo anterior, o sintagma "de laranja" no enunciado 2 corresponde a um constituinte no enunciado 1, a saber, "de banana", e pode ser substituído por este formando uma nova sentença completa e com significado: "Qual foi a área colhida, a área plantada e a produção *de laranja* nos estados da região Nordeste entre os anos de 1970 e 1975?"

A referência cruzada entre o enunciado corrente e os anteriores resulta numa visão das sentenças como entidades interelacionadas, criando uma estrutura de diálogo. Logo, o tratamento de elipses é um dos aspectos que se relaciona com a sensibilidade da linguagem ao contexto. Além disso, o uso de elipses é tão frequente nos diálogos humanos que é importante que as interfaces em LN sejam capazes de lidar com este fenômeno.

O processamento de elipses da interface AGRON é baseado no **PSLI 3** [Frederking, 1986], um sistema de compreensão e geração de LN desenvolvido no Departamento de Ciência da Computação da Universidade de Carnegie-Mellon (Pennsylvania, U.S.A.), objeto da dissertação de doutorado de R. Frederking. O sistema utiliza uma arquitetura baseada em regras de reescrita e uma adaptação da técnica de análise sintática através de *charts* [Kay, 1973; Winograd, 1983] para incluir semântica. A flexibilidade da interface reside no processamento de elipses e de referências definidas estabelecendo uma estrutura de diálogo e permitindo análise contextual.

Frederking desenvolveu uma taxonomia de classificação de enunciados elípticos baseada no tipo de processamento necessário para interpretar a sentença. Num nível inicial, as sentenças elípticas são classificadas segundo a resolução num enunciado antecedente ou em outro método não-linguístico. A elipse é dita *antecedente* se depende de um enunciado prévio, caso contrário é chamada de *não-antecedente*.

Um exemplo de elipse *não-antecedente* pode ser a situação onde algumas pessoas se sentam à mesa de jantar e uma delas pergunta: "Sal?" Neste caso, a sentença elíptica não depende de nenhum enunciado prévio. As elipses *não-antecedentes* ainda são classificadas em dependentes ou não do contexto. Assim, uma resposta "Vazio!" à pergunta anterior é um exemplo de elipse *não-antecedente dependente do contexto*. As elipses *não-antecedentes* não serão tratadas no sistema AGRON, pois de um modo geral não se aplicam ao escopo da interface.

As elipses *antecedentes* são classificadas em *intrasentenciais* e *intersentenciais* dependendo se o antecedente se encontra na sentença corrente ou numa sentença prévia. Como vimos anteriormente, trabalhamos as elipses *intrasentenciais* ao processar as cláusulas relativas do caso semântico de restrições. Porém, aqui daremos ênfase às elipses *intersentenciais*, que são mais interessantes do ponto de vista do processamento de LN. Assim, quando nos referirmos a elipses *antecedentes*, subentende-se aqui as *intersentenciais*.

As elipses *antecedentes* são divididas em dois tipos, *correferenciais* e *não-correferenciais*, de acordo com a referência a mesma instância de uma ação ou objeto físico. A seguir temos um exemplo de enunciados com elipses *antecedentes correferencial* e *não-correferencial*, respectivamente.

Exemplos:
1: "Liste o valor da produção de soja na região Sul."
2: "A área plantada."
3: "E de uva?"

Neste exemplo, a elipse do enunciado 2 se refere ao mesmo objeto, "soja", por isso é dita *correferencial*. No enunciado 3, a elipse é *não-correferencial* porque se refere a um outro objeto de pesquisa, "uva".

Frederking argumenta que, embora muitos sistemas computacionais assumam que cada enunciado se refere a uma nova ação, parece mais razoável supor que dois enunciados sejam correferentes, a menos que haja uma razão para acreditar no contrário. A mudança de objeto ou ação pode se dar, inclusive, de forma explícita através de marcas no enunciado elíptico, tal como o pronome "outro", que implica, sem dúvida, uma referência a um objeto diferente, e consequentemente, uma elipse *não-correferencial*. Assim sendo, dois enunciados que tenham a mesma estrutura de casos, ou estruturas compatíveis, representando instâncias distintas de uma mesma ação ou objeto, devem ser correferentes.

Esta taxonomia é perfeitamente adequada a uma gramática de casos como a do sistema AGRON, porque, no aspecto semântico, os enunciados elípticos representam a substituição ou o acréscimo de casos no enunciado anterior. A substituição do objeto de pesquisa no caso *objeto* equivale a uma elipse *não-correferencial*, como ocorre no enunciado 3 do último exemplo; a substituição ou o acréscimo dos casos *espaço*, *tempo* e *restrição*, e de modificadores no caso *objeto*, correspondem a elipses *correferenciais*, uma vez que o objeto permanece o mesmo, como no enunciado 2 do exemplo.

No caso dos enunciados elípticos *não-correferenciais*, a elipse é dita de *reformulação*, pois embora o objeto ou ação não seja correferente, a sentença herda alguns constituintes da sentença anterior. No caso do AGRON, significa que os modificadores do objeto, além dos outros casos *espaço*, *tempo* e *restrição*, podem se repetir para o novo enunciado.

As elipses *correferenciais* ainda podem ser subdivididas em três espécies: *elaboração*, *correção* e *eco*. As duas últimas não serão tratadas no AGRON porque não são interessantes no contexto do sistema, porém daremos um exemplo ilustrativo de cada uma delas.

Suponha a situação de um freguês conversando com o garçon num restaurante. Os enunciados do freguês serão representados por F*n*, (onde *n* é uma sequência de números para identificar a sentença), e os do garçon por G*n*.

F1: "Por favor, traga-me um bife com fritas."
G2: "Um frango com fritas?"
F3: "Um bife com fritas."
G4: "Um bife com fritas."

Na sequência de enunciados acima, a sentença F3 representa uma elipse *correferencial de correção*, e na sentença G4 ocorre uma elipse *correferencial de eco*. Nestes dois casos, a participação do ouvinte é requisitada, orador e ouvinte estabelecem um diálogo. No caso da interface AGRON, não existe propriamente um diálogo com o usuário, por isso estas classificações não se aplicam.

Por outro lado, a utilização de elipse *correferencial de elaboração* é muito útil no ambiente do sistema AGRON. Seguem-se algumas sentenças que caracterizam este tipo de elipse.

1: "Qual foi a área plantada de tangerina no estado do Rio de Janeiro em 1980?"

2: "E em São Paulo?"

3: "E a área colhida no Rio de Janeiro?"

4: "No Paraná."

Nota-se que no enunciado 2 e no enunciado 4 a elaboração ocorre no caso espacial, enquanto que na sentença 3, a elaboração é simultaneamente nos caso espacial e no modificador do objeto. A instância do caso temporal permaneceu a mesma em todos os enunciados.

Do ponto de vista computacional, mantemos uma estrutura de *slots* de casos na memória, que são preenchidos e ou substituídos de acordo com o tipo de elipse. A dificuldade reside, então, em identificar o caso semântico que corresponde à elipse, o que será resolvido pelo algoritmo de análise, os outros casos são herdados da sentença precedente.

Segundo Frederking, conceitualmente temos quatro passos na análise dos enunciados elípticos:

- reconhecer o fragmento elíptico - o sistema precisa decidir que a sentença é uma elipse, e não uma sentença mal-formada ou com erros ortográficos;
- casar com o enunciado antecedente - o antecedente do fragmento elíptico deve ser identificado. Na aplicação corrente, supõe-se que o antecedente é sempre o último comando;
- instanciar o novo enunciado - o sentença antecedente e a elíptica são combinadas para formar um novo enunciado. Neste momento é onde a resolução baseada em casos é usada;
- determinar o efeito pretendido - determinar o propósito comunicativo do enunciado segundo o contexto conversacional. No âmbito da interface AGRON, o significado pretendido das sentenças é determinado por regras semânticas bem definidas que relacionam a solicitação do usuário às estruturas do banco de dados.

Na prática, o processamento da elipse começa quando o analisador encontra uma sentença que não pode ser interpretada como uma sentença completa. No caso da gramática do AGRON ocorrem três situações onde a sentença é considerada elíptica:

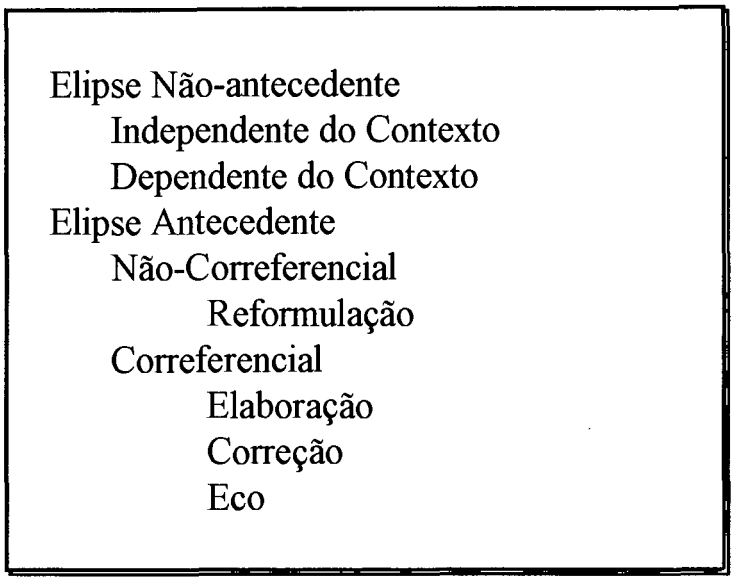
- o enunciado não possui um objeto de pesquisa; assume-se uma elipse de elaboração;
- o enunciado possui um objeto sem modificador; configura-se uma elipse de reformulação;
- o enunciado possui um objeto com modificador, porém a sentença não é iniciada por um verbo imperativo, nem por um pronome interrogativo; também neste caso ocorre uma elipse de reformulação.

No último caso, temos, na verdade, uma marca implícita que denuncia a sentença elíptica, isto é, a ausência dos indicadores de comando imperativo ou pergunta, do contrário não seria possível distinguir o enunciado elíptico de uma outra consulta independente. Considere as sentenças:

1: "Liste a área colhida de trigo no Nordeste em 1989."
2: "Em 1988."
3: "A de cana-de-açúcar"
4: "A área plantada de milho."
5: "Informe a área plantada de milho no Sudeste."

Os enunciados elípticos 2, 3 e 4 correspondem, respectivamente as três situações de elipse descritas anteriormente, e todos *correferem* ao enunciado 1. A última sentença, contudo, é considerada uma pesquisa independente das anteriores, pois é introduzida por um verbo no imperativo caracterizando um novo comando. Desta forma, assumimos que o enunciado 5 *não-correfere* ao enunciado 1.

A taxonomia completa de Frederking para a análise de elipses é dada a seguir:



Em resumo, na interface AGRON serão tratadas, principalmente, dois tipos elipses - elipses *intersentenciais antecedentes não-correferenciais de reformulação* e elipses *intersentenciais antecedentes correferenciais de elaboração* - segundo a taxonomia de Frederking. A elipse de reformulação ocorre na substituição do objeto de pesquisa por outro. No domínio da produção agrícola equivale a referenciar um produto agrícola diferente. A elipse de elaboração adiciona ou substitui os casos complementares ou os modificadores do objeto. Além disso, ocorre a elipse *intrasentencial* nas cláusulas relativas do caso *restrição*.

Exemplo:

"Informe a área plantada de milho nos estados do Sudeste cuja produção é inferior a 50 toneladas."

Reformulação: "E de soja?"
Elaboração: "E com a produção maior que 50 toneladas?"

Duas linhas de pesquisa anteriores foram relevantes no tratamento de elipses. A primeira ocorreu com o sistema LADDER [Hendrix, 1977], e mais tarde um outro enfoque foi desenvolvido no sistema DYPAR [Carbonell, 1983]. Este último é basicamente o mesmo mecanismo do PSLI 3 e também da interface AGRON.

No primeiro sistema, no LADDER, utilizava-se um mecanismo baseado na similaridade sintático-semântica entre a sentença elíptica e a sentença anterior, ou seja, o sistema aceitava qualquer sequência de palavras que fosse sintaticamente análoga a qualquer subsequência de palavras do último enunciado. A técnica

consistia em reconhecer a elipse como um não terminal da gramática semântica, atráves da análise bottom-up da sentença elíptica. Esta método apresentava alguns problemas. O primeiro é que a estratégia bottom-up de análise da elipse diferia da análise das outras sentenças, que era top-down. Além disso, o sistema não conseguia lidar com enunciados elípticos onde a forma sintática do constituinte diferia do correspondente no enunciado anterior.

No DYPAR, assim como no PSLI 3 e na interface AGRON, mesclam-se técnicas de estrutura de casos e gramática semântica, onde o casamento de padrões da gramática semântica é usado para reconhecer a relação temática do caso. Nesta técnica, como dissemos, o enunciado antecedente e o fragmento elíptico são combinados, reescrevendo-se ou adicionando-se os casos presentes na sentença elíptica, e retendo-se os casos do antecedente que não estão no enunciado elíptico.

VII. A GRAMÁTICA

Segundo Allen [1987], uma boa gramática deve considerar três aspectos fundamentais:

- generalidade - diz respeito ao conjunto de sentenças bem-formadas que podem ser analisadas pela gramática;
- seletividade - compreende o conjunto de sentenças mal-formadas, ou não-sentenças, que devem ser rejeitadas pela gramática;
- compreensibilidade - reflete o grau de simplicidade da gramática.

Para se verificar a aplicabilidade destes três princípios numa gramática, existem alguns testes que podem ser efetuados. Um procedimento adequado consiste em, ao identificar um constituinte gramatical, verificar a construção de uma sentença utilizando a conjunção deste constituinte com outro do mesmo tipo, porque, numa sentença gramaticalmente correta, somente constituintes do mesmo tipo podem ser ligados por conjunção. Se a gramática permite a conjunção de constituintes de tipo diferente, então ela possui um poder seletivo fraco. Por exemplo, considere o seguinte sintagma:

"a produção de açúcar e no ano de 1975"

A gramática não deve permitir esta construção mal-formada, pois ocorrem dois tipos diferentes de sintagma: o primeiro, "a produção de açucar" é um sintagma nonimal, ao passo que "no ano de 1975" é um sintagma preposicional.

Além disso, pode-se checar o grau de adequação da gramática através da observação do desempenho do sistema. Porém, os resultados assim obtidos não garantem a eficiência da teoria subliminar. O conjunto de sentenças submetidas aos testes podem, por exemplo, não refletir uma variedade suficientemente grande de sentenças necessárias para sustentar um nível razoável de generalidade e seletividade da gramática.

De um modo geral, rodar o programa e observar a sua performance tem sido o método mais utilizado para avaliação de modelos computacionais da linguagem. Contudo, este pode ser um procedimento tendencioso, como se verifica no caso conhecido do sistema ELIZA [Weizenbaum, 1966]. Na verdade, este programa tem pouca teoria linguística subjacente, no entanto o seu desempenho é impressionante, parecendo efetivamente "inteligente", devido a tendência que o próprio usuário tem de atribuir significado ao que o sistema diz.

A linguística computacional precisa de métodos comprovadamente eficazes de avaliação, o que tem sido a preocupação e o objeto de estudo de muitos pesquisadores, especialmente dos lógicos. Entretanto, neste trabalho o nosso objetivo é de elaborar e/ou avaliar técnicas computacionais de processamento da Linguagem Natural que se apliquem ao desenvolvimento de interfaces para bancos de dados. Assim, apesar das possíveis inconsistências e incorreções, nos limitaremos a uma análise exploratória da performance da interface baseada numa série de testes.

Gramáticas Semânticas

A gramática utilizada na definição da linguagem compreendida pelo AGRON é uma gramática semântica de casos. Primeiramente, é uma gramática semântica porque os constituintes dependem do significado das palavras, e não do papel sintático que representam na sentença. Em segundo lugar, é uma gramática de casos porque busca identificar estruturas que preencham os casos semânticos definidos para contemplar as aplicações da linguagem.

A razão principal para se codificar informação semântica na gramática é eliminar, o quanto antes, interpretações sintáticas que não se aplicam semanticamente, além da consequente redução na ocorrência de ambiguidades.

A sentença "Listar a produção de cana-de-açúcar no estado de Fortaleza" é uma sentença sintaticamente correta do ponto de vista de concordâncias, formação de sintagmas nominais e verbais, entre outros aspectos. Porém, não existe um significado conhecido para o SN "estado de Fortaleza", simplesmente porque não existe tal estado. Contudo, uma gramática voltada para a sintaxe gastaria tempo para produzir uma árvore sintática para uma tal sentença, e somente uma análise semântica posterior seria capaz de apontar o erro.

Além disso, numa gramática essencialmente voltada para a sintaxe, a análise gramatical pode resultar numa sentença sintaticamente ambígua, que deixa de sê-lo quando analisada do ponto de vista semântico. Considere a sentença:

"Qual a produção de castanha do Pará em 1980?"

Suponhamos uma ambiguidade estrutural no sintagma preposicional "do Pará", pois este pode estar qualificando o substantivo "castanha" para formar o SN "castanha do Pará", ou especificando o local onde se deseja buscar a produção de castanha, isto é, no Pará.

Porém, no domínio semântico do banco de dados BMA não existe informação sobre um produto chamado "castanha do Pará", mas somente sobre um outro tipo de castanha, "castanha de cajú". Logo, para o AGRON, esta sentença não pode ser analisada. Neste caso, a ambiguidade foi resolvida a nível semântico, pois não existe um significado para o sintagma "castanha do Pará" no domínio da aplicação. Retomaremos a discussão sobre esta sentença ao tratarmos das preferências semânticas incorporadas pelo analisador.

As gramáticas semânticas tendem a ter um número de regras superior as correpondentes gramáticas sintáticas, porque, em geral, o que ocorre é o desmembramento de uma regra em duas ou mais regras que especificam as instâncias dos constituintes gramaticais que são semanticamente válidos.

Seja um subconjunto de regras de uma gramática sintática para formação de um sintagma verbal composto de verbo, objeto direto e complementos:

SV → VERBO SN SP
SV → VERBO SN
SN → ART SUBST ADJT
SN → SN SP
SP → PREP SUBST

A gramática semântica correspondente para contemplar um determinado domínio semântico tem de especificar quais os tipos de SN e SP que são aceitos por cada verbo a ser processado.

Suponhamos o domínio semântico da Produção Agrícola no Brasil. Assim, um subconjunto de regras semânticas para os comandos imperativos que listam os atributos de um produto agrícola em um estado poderia ser:

SV → COMD OBJT ESPC
SV → COMD OBJT
OBJT → LMOD OBJS
LMOD → DETM MODF LMODF
LMOD → DETM MODF
ESPC → SPRE UNFD

onde os não terminais COMD, OBJS, MODF, DETM, SPRE e UNFD representam *classes semânticas* de terminais, nesta ordem, classe dos verbos de comando imperativo, objetos (correspondem aos produtos agrícolas), modificadores do objeto, determinadores (artigos, quantificadores e numerais cardinais), preposições que indicam situação (do, da, no, na ...) e unidades da federação. Os não-terminais OBJT e ESPC representam os casos *objeto* e *espaço*, respectivamente.

Neste caso, porém, não se verificou um aumento significativo no número de regras gramaticais devido ao artifício de criarmos *classes semânticas*, isto é, o agrupamento de terminais em classes bem definidas segundo a semântica do domínio de aplicação. Por exemplo, a classes dos objetos simples no AGRON é, caracteristicamente, o conjunto dos produtos agrícolas pesquisados. Da mesma forma, o conjunto de estados brasileiros constitui a classe das unidades da federação. E assim também ocorre com a divisão das preposições em classes, de acordo com o

seu papel no universo semântico.

As gramáticas semânticas têm sido usadas em diversos sistemas para processamento de Linguagem Natural. Dentre os mais conhecidos podemos citar LIFER [Hendrix et al., 1978] e PLANES [Waltz & Goodman, 1977].

Segundo Allen, as gramáticas semânticas, na prática, são extremamente úteis para produzir uma interface em LN robusta para domínios limitados onde é possível analisar manualmente o conjunto de todas as possíveis questões. No entanto, quando se trata de sistemas mais gerais, a tarefa de construir gramáticas semânticas torna-se difícil, a menos que se tenha algum tipo de geração automática da gramática a partir de informação sintático-semântica especificada pelo usuário.

Logo, por um lado, a vantagem maior de se ter sintaxe e semântica em processos separados é permitir modelos de análise mais abrangentes porque a parte sintática do sistema é relativamente independente da porção semântica, aumentando a generalidade da gramática. Por outro lado, a combinação de sintaxe e semântica num único estágio, tal como nas gramáticas semânticas, permite o uso simultâneo de informação semântica para eliminar construções sintáticas que são anômalas semanticamente.

Gramáticas de Casos

O formalismo das relações de *casos* foi introduzido por Fillmore [1968] para descrever certas relações entre verbos, sintagmas nominais e sintagmas preposicionais numa sentença. O objetivo era estabelecer generalizações linguísticas de como o verbo exige determinados complementos ou *casos* dependendo da sua função semântica.

Em linguística, este paradigma deu origem às chamadas gramáticas de casos que, em suma, descrevem o conjunto de papéis semânticos, ditos *casos*, que os sintagmas nominais e preposicionais podem desempenhar junto aos verbos aos quais estão ligados. Mesmo não havendo um consenso por parte dos pesquisadores de quantas e quais exatamente devem ser estas relações semânticas, existem alguns casos bem definidos que podem ser observados em muitas situações.

Embora haja uma opinião geral de que o número de casos semânticos é relativamente pequeno, existem diferentes estruturas de casos propostas por diferentes estudiosos. Isto se deve ao fato de que cada estrutura de casos focaliza um aspecto e um nível de hierarquia semântica diferenciado, e obviamente, cada um defende que a sua visão pode ser a mais genérica e aplicável.

Divergências a parte, existem certas estruturas típicas, tais como os casos:

- *agente* - aquele que causa um evento intencionalmente;
- *tema* - aquilo que é afetado pelo evento; geralmente corresponde ao objeto direto sintático de um verbo transitivo direto, ou ao sujeito que não é agente no caso de um verbo intransitivo;
- *instrumento* - instrumento ou força usada para executar um evento; forças naturais também são incluídas neste caso;
- *experimentador* - o indivíduo que percebe algum evento, ou acredita que ele ocorre, caracterizando um estado psicológico específico; o experimentador não é propriamente um agente porque nem sempre existe intencionalidade no estado experimentado;
- *beneficiário* - a pessoa para a qual algum evento ou ato é dirigido;
- *temporal* (no tempo) - época ou momento de ocorrência de um evento;
- *local* (no local) - local ou dimensão onde o evento ocorre;
- *local-fonte* (do local) - local original do evento;
- *local-destino* (para o local) - local final do evento.

Uma vez que o objetivo final na análise de casos é fornecer o significado da sentença, diferentes estruturas sintáticas podem mapear uma mesma estrutura ou uma estrutura de casos semelhante. Sem dúvida, a inversa também é verdadeira, ou seja,

sentenças com significados diferentes, embora com estruturas sintáticas semelhantes, mapeam resultados distintos. Na verdade, toda a análise gramatical que incorpora alguma semântica busca encontrar uma espécie de estrutura profunda, do tipo definido por Chomsky, relativamente independente da estrutura superficial da sentença apresentada.

Um exemplo clássico de mapeamento numa estrutura de casos é o conjunto de sentenças:

"João quebrou a janela com o martelo."
"O martelo quebrou a janela."
"A janela quebrou."

Em todas as sentenças, apesar das variações sintáticas, os casos *agente*, *tema* e *instrumento* são os mesmos: João é o agente, o tema é a janela, e o instrumento é o martelo.

Esta é a teoria subjacente à definição dos casos *objeto*, *espaço* e *tempo* na interface AGRON, que correspondem, até certo ponto, aos casos *tema*, *local* e *espacial*, respectivamente. Além destes, no sistema AGRON temos o caso *restrição* que é mais específico ao problema de limitar o universo de busca num banco de dados.

É interessante observar que, embora o caso *agente* seja um dos casos principais em muitos formalismos gramaticais, geralmente presente em todo o tipo de assertiva, não é um caso analisado no sistema AGRON. Na prática isto ocorre simplesmente porque o objetivo do sistema é o processamento de perguntas e comandos imperativos, logo a gramática não é voltada para a análise de assertivas e, consequentemente, não considera o caso *agente*.

O que se pretende, com a maneira como a gramática do AGRON foi definida, é poder, independentemente da forma sintática superficial da sentença, identificar o objeto de pesquisa (geralmente um produto agrícola), além de, quando presentes, um local para consulta (um conjunto de municípios, estados ou regiões do Brasil), uma determinada época (um ano ou conjunto de anos de coleta da informação), e um conjunto de restrições de busca.

Classificação de Verbos

Considerados no prisma das relações semânticas de casos, os verbos podem ser classificados numa hierarquia segundo os casos que requerem. A classificação de verbos tem sido objeto de muitas pesquisas linguísticas, na área computacional e em outras, como acontece na classificação utilizada pelo modelo de representação baseado em redes semânticas de Schank [Schank, 1973; Schank e Riesbeck, 1981] conhecido como "dependência conceitual". No modelo de Schank, além de um número limitado de casos verbais, existem as chamadas ações "primitivas", que combinadas entre si dão origem aos outros verbos da hierarquia.

O trabalho de Schank tem influenciado muitos outros pesquisadores, e serve para apontar uma tendência da linguística computacional: a utilização da estrutura de *casos* em muitas representações de Inteligência Artificial, especialmente na análise de Linguagem Natural.

Existe também a especulação, discutida por Charniak [1981], de que a teoria de representação baseada em *frames* [Minsky, 1975] e as gramáticas de casos são formalismos equivalentes, uma vez que a estrutura de *slots* utilizada nos *frames* pode corresponder a estrutura de casos da gramática de casos.

Na interface AGRON a classificação de verbos é secundária, porque utilizamos a suposição de que lidamos com um conjunto de objetos, que têm importância central, armazenados segundo dados históricos, isto é, com um caráter temporal fundamental, além da observação do aspecto espacial da informação. Logo, qualquer verbo semanticamente importante ligado ao domínio semântico terá sempre uma função espaço-tempo em relação a um objeto ou tema. Por outro lado, na classificação de verbos é a própria ação que tem a ênfase principal.

Na verdade, o que importa aqui é definir uma estrutura semântica espaço-temporal que sirva de modelo para a consulta de informações sobre determinados objetos num banco de dados. Neste caso, podemos supor que, em todos os bancos de dados sobre um determinado tema, resultantes de pesquisas com uma certa periodicidade, realizadas em um espaço qualquer, haverá um conjunto de informações que sempre irá preencher os casos *objeto*, *espaço* e *tempo* definidos na gramática. A visão espaço-temporal de um objeto de pesquisa é o suporte de uma extensão para a independência do domínio de aplicação, permitindo um certo nível de transportabilidade da interface AGRON.

A Gramática do AGRON

A princípio, a gramática do AGRON é uma gramática livre de contexto, composta de 70 regras descritas no apêndice A. Porém, a análise das sentenças permite o processamento de elipses, caracterizando, desta forma, a sensibilidade da gramática ao contexto, através de um conjunto de construções elípticas adicionais que não estão explícitas nas regras. Além disso, como vimos, utilizaremos uma gramática semântica de casos, que aceita comandos imperativos e perguntas, com quatro casos básicos (*objeto*, *espaço*, *tempo* e *restrição*), voltada para um domínio semântico produto-agrícola-no-espaço-tempo, onde o espaço se caracteriza por municípios, estados ou regiões, e o tempo é medido em anos.

A notação é a mesma notação comumente usada para descrever as regras gramaticais ($A \rightarrow B$), segundo a definição que apresentamos anteriormente, com a ressalva de que teremos somente a especificação de símbolos não-terminais. Utilizaremos sempre uma abreviatura de quatro letras para cada não-terminal, e o símbolo inicial será representado por S.

A diferença, neste caso, é que alguns não-terminais, aqueles que definem as chamadas *classes semânticas*, podem ser diretamente mapeados em terminais (palavras da linguagem) através de informações armazenadas no léxico; ao passo que os outros não-terminais representam casos semânticos ou constituintes gramaticais mais complexos, e precisam ser submetidos às regras da gramática antes de poderem ser convertidos nos primeiros.

No primeiro nível, a sentença é dividida em dois tipos de não-terminais: indicadores de comando ou interrogativas (comd ou itrr) mais uma sequência de casos semânticos correspondentes aos quatro casos básicos *objeto*, *espaço*, *tempo* e *restrição* (objt, espc, temp, rest). Além disso, pode ou não haver uma marca explícita, a conjunção *e*, indicando que o enunciado é uma sentença elíptica.

Num segundo desdobramento, trabalhamos as construções possíveis para cada caso semântico. O caso *espaço* possui três níveis de aplicação: o espaço municipal (emun), o espaço estadual ou de unidade da federação (eunf) e o espaço regional (ereg), indicando, respectivamente quando a referência é a municípios, estados ou regiões do Brasil. Para cada tipo de espaço existe um conjunto de constituintes e construções específicas, dando origem a um conjunto de regras específicas que considera a hierarquia de tipos e as relações de pertinência discutidas na seção passada sobre o caso espacial e suas características.

O caso *tempo* (temp) consiste na especificação de uma lista ou um intervalo de anos, que podem ser introduzidos por certas preposições e locuções prepositivas, classificadas como preposições que marcam uma condição temporal (cpre), tais

como *além de*, *antes de*, *após*, *depois de*, *desde*, *entre*. Outras duas classes de preposições podem introduzir o caso temporal, a preposição *para* (ppre) e as preposições ditas de situação (spre) *do*, *dos*, *no*, *nos*. A classificação das preposições será fornecida no capítulo sobre o léxico.

O caso *objeto* é o mais complexo. Os objetos são, tipicamente, produtos agrícolas. Contudo, existe a situação onde se deseja obter os municípios, estados ou regiões que obedecem certas características, bem como os anos onde ocorreu um conjunto de condições. Logo, os objetos, na verdade, podem ser de três tipos: produtos agrícolas, um local ou uma época.

Quando o objeto é simples, ocorre a referência a um e somente um produto agrícola. Porém, pode ocorrer a especificação de um objeto composto, neste caso, um tipo de lavoura (temporária ou permanente) que caracteriza uma forma de generalização dos produtos agrícolas, segundo exposto na seção sobre o caso *objeto* e a hierarquia de subsunções. Além disso, os objetos podem possuir modificadores, tais como "a área colhida" e "o valor da produção".

Se o objeto é uma referência a um local, pode ocorrer a qualificação do objeto, por exemplo, quando dizemos "os estados produtores". Destacamos este tipo de qualificação dos modificadores por tratar-se de uma resolução semântica diferente: os modificadores correspondem a atributos do objeto que são obtidos diretamente através de campos ou colunas numa tabela do banco de dados; os qualificadores são associados a ações semânticas, e estão condicionalmente ligados a campos do banco de dados. Por exemplo, para o qualificador "produtores" está associada uma regra semântica que é a mesma do verbo "produz", ou seja, um município, estado ou região *produz* ou é dito *produtor* quando a produção daquele determinado produto é maior que zero.

O caso *restrição* contempla três tipos de situação:

- a condição não possui um verbo;
- possui um verbo de ligação;
- possui um verbo de ação ligado ao domínio semântico.

Exemplos destas três formas são:

"com o valor da produção menor que Cr$ 10.000.000,00"
"cuja área colhida é superior a 100 hectares"
"que produzem mais de 15 toneladas de soja"

Além disso, pode haver conjunção de condições restritivas como em:

"cuja produção foi maior que 100 toneladas e inferior a 200 toneladas".

Classes Semânticas e Não Terminais

É importante ressaltar que, contrariamente ao aumento no número de regras que geralmente é esperado na geração de uma gramática semântica, a utilização de classes semânticas foi um recurso que permitiu uma redução no número de regras gramaticais, uma vez que não é necessário repetir a regra para elementos da mesma classe.

Por exemplo, considere a seguinte regra semântica associada ao caso espacial da gramática que especifica o espaço das unidades da federação:

espc → spre unfd

A regra não precisa ser recodificada para cada uma das preposições que integram a classe *spre*, neste caso, as preposições *da*, *das*, *do*, *dos*, *na*, *nas*, *no*, *nos*, *em*. Desta forma, a regra contempla igualmente sintagmas tais como:

"no Espírito Santo"
"em São Paulo"
"da Bahia"

A dinâmica do uso de classes semânticas está intimamente associada ao funcionamento do léxico. Ou seja, existe uma especificação de classe associada a cada entrada léxica, permitindo assim o agrupamento das palavras segundo um critério semântico. Na verdade, a especificação da classe semântica diretamente no léxico substitui, até certo ponto, o uso de representações do conhecimento mais sofisticadas para codificar semântica, como redes semânticas. Neste caso, a informação da classe semântica no léxico pode ser vista, por exemplo, como um elo *isa* entre a classe e a instâncias que a compõem.

Considere, a seguir, a relação das trinta e uma classes semânticas definidas para a implementação da interface AGRON. Após cada ítem, segue-se o exemplo de um sintagma que contém uma palavra (em itálico) correspondente a classe definida.

- sobj - identifica o substantivo "produto" ou seus sinônimos
  Ex: "os *produtos* da região Norte"
- sano - identifica o substantivo "ano" ou seus sinônimos
  Ex: "no *ano* de 1980"
- smun - identifica o substantivo "município" ou seus sinônimos
  Ex: "do *município* de Salvador"
- sunf - identifica o substantivo "unidade da federação" ou seus sinônimos
  Ex: "no *estado* do Maranhão"
- sreg - identifica o substantivo "região" ou seus sinônimos

Ex: "da *região* Sudeste"

- bras - identifica o substantivo "Brasil"
  Ex: "no *Brasil*"
- munc - referencia o nome de um município
  Ex: "em *Niterói*"
- unfd - referencia o nome de uma unidade da federação
  Ex: "em *Goiás*"
- regi - referencia o nome de uma região
  Ex: "do *Nordeste*"
- objs - define um objeto simples, neste caso, o nome de um produto agrícola
  Ex: "produção de *soja*"
- objc - define um objeto composto, neste caso, os produtos de um mesmo tipo de lavoura
  Ex: "produtos da *lavoura temporária*"
- modf - identifica um modificador que antecede o objeto
  Ex: "*valor da produção* de milho"
- qual - identifica um adjetivo qualificador derivado de um verbo
  Ex: "estados *produtores* de uva"
- cpre - aponta uma preposição que marca condição temporal
  Ex: "*antes* de 1984"
- dpre - aponta uma preposição que marca deslocamento temporal
  Ex: "de 1975 *até* 1980"
- spre - aponta uma preposição que marca situação
  Ex: "*no* estado de Alagoas"
- qpre - aponta uma preposição ou locução prepositiva de quantidade
  Ex: "produção *igual a* 10000 cachos"
- mpre - aponta uma preposição ou locução prepositiva que marca medida
  Ex: "estado que produz *mais de* 100 toneladas"
- ppre - indica a preposição "para"
  Ex: "*para* os municípios da Paraíba"
- artd - identifica artigos definidos
  Ex: "*o* valor da produção"
- quat - define os quantificadores, ou seja, o conjunto de pronomes indefinidos
  Ex: "para *cada* estado da região Norte"
- card - aponta um numeral cardinal
  Ex: "*dois* estados"
- cond - introduz as condições das cláusulas restritivas, ou seja, define o conjunto de pronomes relativos
  Ex: "*cuja* quantidade produzida foi maior que 100 toneladas"
- numr - indica um número ou quantidade
  Ex: "*1000* hectares"
- cjun - define a conjunção aditiva "e"
  Ex: "entre 100 *e* 150 hectares"

- djun - define a conjunção alternativa "ou"
  Ex: "nos anos de 1982 *ou* 1983"
- unmd - aponta uma unidade de medida
  Ex: "100 *toneladas*"
- depn - referencia uma dependência semântica, ou seja, uma palavra cujo significado complementa e/ou depende do significado de outra, e, por isso, não tem valor na análise da sentença, podendo ser ignorada; compreendem adjetivos ou expressões adjetivas, os verbos auxiliares "ser" e "ter" e a preposição "de"
  Ex: "arroz *sequeiro*"
- itrr - identifica uma interrogação ou perguntas *qu*, isto é, compreende os pronomes e advérbios interrogativos
  Ex: "*Qual* a área colhida ..."
- comd - identifica um comando imperativo, ou seja, compreende os verbos "exibir", "mostrar", "listar", e outros
  Ex: "*Liste* os produtos ..."
- verb - identifica um verbo específico do domínio semântico tais como "produzir", "plantar", "coletar".
  Ex: "os estados que *produzem* feijão"

Além das classes semânticas, temos também o conjunto de não-terminais que, como dissemos, são utilizados para denominar os casos e outros constituintes compostos da gramática. Para diferenciar das classes semânticas, estes não-terminais serão especificados entre colchetes no apêndice A que contém a descrição das regras gramaticais. A seguir damos a lista destes quatorze não-terminais. Segue-se, igualmente, um exemplo de um sintagma equivalente.

- objt - caracteriza o caso objeto
  Ex: "o valor da produção de cebola"
- lmod - lista de modificadores
  Ex: "a área plantada e a área colhida"
- detm - compreende os tipos de determinadores dos objetos, neste caso, artigos definidos, pronomes indefinidos ou numerais cardinais
  Exs: "os", "cada", "dois"
- espc - caracteriza o caso espacial em geral
  Ex: "na região Norte"
- emun - expressão que identifica um espaço a nível municipal
  Ex: "em Fortaleza"
- eunf - expressão que identifica um espaço a nível de unidade da federação
  Ex: "nos estados da região Sul"
- ereg - expressão que identifica um espaço a nível de região
  Ex: "no Nordeste"
- lmun - lista de municípios

Ex: "Curitiba, Florianópolis e São Paulo"

- lunf - lista de unidades da federação
  Ex: "Amazonas e Pará"
- lreg - lista de regiões
  Ex: "Sul e Sudeste"
- temp - caracteriza o caso temporal
  Ex: "em 1980"
- lano - lista de anos
  Ex: "1975, 1976 e 1977"
- rest - caracteriza o caso restritivo
  Ex: "com a produção superior a 15000 cachos"
- lres - lista de condições restritivas
  Ex: "maior que 100 e menor que 200 toneladas"

VIII. ANÁLISE LÉXICA

O analisador léxico, também conhecido como *scan*, constitui a fase inicial de análise no processamento das sentenças. A análise léxica é uma análise linear onde os caracteres que compõem a entrada do sistema são lidos da esquerda para a direita e agrupados a fim de formar os terminais ou lexemas da linguagem. Na verdade, a divisão entre análise léxica e análise sintática é um tanto arbitrária. Geralmente, o fator que determina a divisão é a recursão: a construção léxica não requer recursão, enquanto a construção sintática requer.

O processo de recuperação de todas as informações sobre uma palavra, incluindo os significados e as características sintáticas, além dos códigos fonológicos e ortográficos, é chamado acesso léxico. Em muitas interfaces em LN, assim como no AGRON e no TEAM, o analisador léxico faz uso de dois dicionários para efetuar o acesso léxico: um dicionário fixo de ítens fechados que independem do domínio da aplicação; e um dicionário móvel de ítens abertos, específico para o domínio em questão. A rigor, o dicionário fixo relaciona os morfemas gramaticais, ou gramemas, enquanto que o dicionário móvel é composto dos morfemas lexicais, ou lexemas propriamente ditos. Porém, para facilitar, usaremos apenas a denominação mais popular de *lexema* para referenciar ambas as situações.

Além disso, o analisador léxico mantém outras informações complementares sobre os lexemas, tais como: se o lexema constitui palavra relevante ou pode ser ignorada na análise; qual o número, pessoa e gênero gramatical; se o lexema está associado a um lexema mais geral ou a um sintagma; qual o atributo da relação do banco de dados; qual o não-terminal da gramática associado; se o lexema pode ser usado para resolver algum tipo de ambiguidade.

A análise léxica aponta um erro quando o lexema não é encontrado nos dicionários. Neste caso, pode ter havido um erro ortográfico ou então a palavra realmente não consta. Então, o sistema emite uma mensagem avisando ao usuário sobre o vocábulo não processado, e qual a dificuldade encontrada na análise da sentença.

O Dicionário Fixo do AGRON

Integram o dicionário fixo da linguagem todos os morfemas gramaticais, ou gramemas, que têm significação interna, isto é, derivam das relações e categorias levadas em conta pela língua, compondo uma série fechada, de número definido e restrito no idioma. São gramemas os artigos, os pronomes, os numerais, as preposições, as conjunções e os advérbios, bem como as formas indicadoras de número, gênero, tempo, modo ou aspecto verbal, excetuando-se, porém, os advérbios de modo.

No caso do AGRON, as considerações sobre cada classe de gramemas são as seguintes:

- Artigos
  Serão tratados os artigos definidos, uma vez que a ocorrência de referências definidas no acesso a banco de dados estatísticos é predominantemente maior do que as referências indefinidas, e contempla a grande maioria das aplicações da interface AGRON.

- Numerais
  Processaremos os numerais cardinais. Não serão considerados os numerais ordinais, multiplicativos ou fracionários. Para reduzir o tamanho da tabela no protótipo, armazenamos somente os numerais cardinais da primeira dezena.

- Pronomes
  Serão considerados pronomes relativos, interrogativos e alguns pronomes indefinidos. Os pronomes pessoais, possessivos e demonstrativos não serão processados.

- Advérbios
  Processaremos advérbios interrogativos, exceto os advérbios *por que* e *como*, devido aos motivos já apresentados. Dispensaremos também o tratamento de advérbios de afirmação, dúvida, intensidade, lugar, modo, negação, tempo, e outros, em virtude do caráter não-declarativo das sentenças.

- Preposições
  Serão processadas preposições e locuções prepositivas segundo a divisão em seis classes criadas para atender as características semânticas da aplicação do sistema, a saber:
  - cpre - indicam uma condição temporal
  - dpre - indicam um deslocamento temporal
  - spre - indicam uma situação particular

- qpre - indicam quantidade, equivalem aos operadores relacionais
- mpre - indicam uma medida de comparação, semelhante a qpre
- ppre - indica a preposição *para*

Não serão tratadas quaisquer locuções prepositivas com mais de três palavras. O limite é apenas uma restrição de implementação que pode ser ampliado conforme as necessidades da aplicação.

- Conjunções
  Serão consideradas somente as conjunções coordenativas aditiva *e* e alternativa *ou*.

- Desinências de gênero e número
  As marcas indicadoras de gênero e número, além do tempo e modo verbal não serão analisadas em separado, isto é, não tratamos aspectos morfológicos. O léxico possui uma entrada para cada variação do radical das palavras, ou seja, temos entradas individuais para as combinações das formas masculina, feminina, singular, plural e conjugações verbais. Isto acarreta um aumento nas entradas dos dicionários, porém ganhamos em eficiência e tempo de acesso. Além disso, a ênfase na análise das palavras na interface AGRON não é voltada para a forma sintática.

No apêndice B consta a tabela de gramemas, juntamente com a classe gramatical, o gênero e o número associados, que compõem o dicionário fixo na implementação do protótipo do projeto AGRON.

O Dicionário Móvel do AGRON

Consideramos como integrantes do dicionário móvel da linguagem todos os morfemas lexicais, ou lexemas, que têm significação externa, isto é, referem-se a fatos do mundo extralinguístico, aos símbolos básicos de tudo o que os falantes distinguem na realidade objetiva ou subjetiva, compondo uma classe aberta de número elevado, indefinido, sempre passível de ser acrescida de novos elementos. São lexemas os substantivos, os adjetivos, os verbos e advérbios de modo. Para simplificar, incluiremos também na relação de lexemas as correpondentes desinências de número, gênero, tempo e modo verbal, que, a rigor, são morfemas gramaticais.

Tendo em vista o domínio semântico agrícola, definimos um conjunto particular de substantivos, adjetivos e verbos, segundo as informações do banco de dados BMA. Devido a essência não-declarativa dos enunciados, não processaremos advérbios de modo. Não existem referências isoladas a adjetivos, pois estes são considerados juntamente com o substantivo que qualificam, caracterizam ou delimitam, formando uma mesma entrada léxica.

Os verbos e expressões verbais serão considerados na terceira pessoa do singular e do plural, nos tempos presente e pretérito perfeito e imperfeito do indicativo, pelas razões já citadas. Exceções a esta regra são os verbos característicos de comandos imperativos, tais como *mostrar*, *exibir*, *fornecer*, entre outros, que serão tratados essencialmente no infinitivo ou no imperativo. Os verbos de ligação e auxiliares compreendem os verbos *ser* e *ter*. Alguns dos verbos específicos do domínio de aplicação são *plantar*, *cultivar* e *produzir*.

Dispensaremos a especificação dos substantivos que correspondem aos nomes de município, estado e região para não sobrecarregar aqui a tabela do dicionário móvel. Na implementação do AGRON, consideramos as cinco regiões, vinte e sete unidades da federação, além de vinte e seis capitais, embora seja apenas uma questão de aumento no tamanho da tabela do programa a extensão para todos os municípios brasileiros, que hoje são cerca de cinco mil.

No apêndice C temos a tabela que constitui o dicionário móvel do protótipo da interface AGRON.

O Analisador Léxico do AGRON

A função do analisador léxico é ler os caracteres da sentença de entrada, agrupá-los em lexemas e passar as palavras assim formadas pelos lexemas, juntamente com as outras informações léxicas, aos outros estágios seguintes de análise, neste caso, ao analisador gramatical. A interação entre as análises gramatical e léxica na implementação da interface AGRON é feita transformando o analisador léxico numa subrotina do analisador gramatical que retorna uma entrada léxica sempre que necessário.

Como vimos, o léxico é dividido em dois dicionários distintos: o fixo e o móvel. Cada entrada léxica nestes dicionários possui oito campos com informações sobre o vocábulo, a saber:

- **pal**: cadeia de caracteres contendo a palavra;
- **clas**: indicador da classe gramatical;
- **gen**: indicador do gênero;
- **num**: indicador do número;
- **nterm**: não-terminal da gramática associado;
- **nexp**: número de palavras adicionais - uma ou duas - com as quais a palavra corrente forma uma expressão;
- **rpal**: conjunto de até duas cadeias de caracteres ou palavras que formam a expressão segundo indicado no campo anterior (nexp);
- **varbd**: associa uma variável ou atributo numa relação do banco de dados, bem como uma expressão que indica possíveis valores para suas instâncias;

Na verdade, somente cinco destes campos - *pal*, *nterm*, *nexp*, *rpal* e *varbd* - têm informação relevante para a análise da sentença na atual implementação do AGRON, sendo que o último deles - *varbd* - não é preenchido para o dicionário fixo. Os outros campos - *clas*, *gen* e *num* - possuem apenas informações de caráter sintático que terão utilidade quando for implementado, por exemplo, um cheque para concordâncias.

Para a expressão "batata doce" temos a seguinte entrada léxica:

```
pal   := 'batata';
clas  := subs;
gen   := f;
num   := s;
nterm := objs;
nexp  := 1;
rpal  := 'doce';
varbd := 'produto',' = 0160';
```

Todas as palavras no léxico são armazenadas em letras minúsculas, sem acentos ortográficos, e sem cedilha. Por exemplo, a palavra *maçã* se transforma em *maca* para o léxico. As únicas exceções a esta regra são os pares de vocábulos *são* e *São*, *para* e *Pará*, que por mapearem a mesma sequência de caracteres, têm a ortografia diferenciada nas entradas léxicas: *sao* e *Sao*, *para* e *Para*, respectivamente.

O analisador léxico possui seis subrotinas:

- PulaSeparador - pula os separadores de palavras - , . ; : ! ? e brancos;
- LeNum - lê um numeral na frase;
- LePal - lê uma palavra na frase;
- ProcuraPal - procura a palavra e devolve a definição léxica;
- Ambiguidade - resolve ambiguidades léxicas;
- ExpSemant - analisa expressão do domínio semântico, decidindo qual a entrada léxica correta que corresponde a expressão.

O algoritmo do analisador léxico é dado a seguir:

```
Procedimento Scan

{ limpa variáveis }
PulaSeparador
Se não é fim de frase
então
        Se o caracter é numérico
        então LeNum
        senão
                LePal
                ProcuraPal
                Se não achou palavra
                então Emite mensagem de erro
                senão
                        Se a palavra é ambígua
                        então Ambiguidade
                        senão
                                Se a palavra compõe expressão
                                então ExpSemant
                                senão
                                        Se é dependência semântica
                                        então Scan
fim
```

IX. ANÁLISE SINTÁTICA

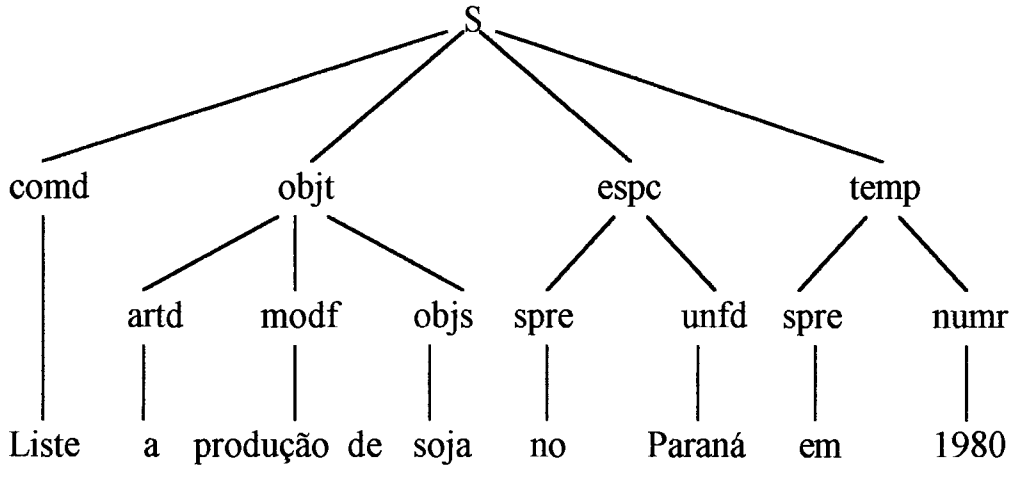
O analisador sintático é o módulo de processamento que analisa as propriedades estruturais da sentença de acordo com as especificações da gramática, produzindo uma representação sintática composta de *slots*, que correspondem aos constituintes gramaticais, e valores para estes *slots*, que correspondem aos terminais da linguagem.

Na interface AGRON, existem dois níveis principais de constituição da sentença. Como vimos na descrição das regras gramaticais, o primeiro nível compõe a estrutura de casos *objeto*, *espaço*, *tempo* e *restrição*. Para cada um destes casos, o segundo nível discrimina os constituintes que são aceitos nas diversas construções, conforme a função semântica que desempenham.

Árvore Sintática

Uma forma bastante comum de representar a estrutura da sentença é através da chamada árvore sintática, que consiste num esquema hierárquico de como os constituintes mais elementares se combinam para formar os constituintes mais complexos, e assim sucessivamente até compor, totalmente, a sentença. Na raiz da árvore é representado o símbolo inicial S da gramática, as folhas da árvore correspondem aos terminais da gramática, e os outros elementos são não-terminais que equivalem aos demais constituintes. A análise sintática também pode ser vista como um processo para encontrar a árvore sintática da sentença.

Considerando a gramática semântica de casos definida para a linguagem do AGRON, temos a seguir a árvore sintática da sentença "Liste a produção de soja no Paraná em 1980."



Analisadores Top-down e Bottom-up

Muitos analisadores utilizam um algoritmo que funciona segundo uma técnica específica de busca para determinar a ordem em que os constituintes são construídos a partir das regras da gramática e da sentença de entrada. Duas técnicas bastante conhecidas são as análises *top-down* (de cima para baixo) e *bottom-up* (de baixo para cima) das sentenças.

Na primeira técnica a construção dos constituintes começa *de cima para baixo* com o símbolo inicial (S) da gramática na raiz da árvore sintática e prossegue em direção aos terminais nas folhas; enquanto que na segunda técnica ocorre o inverso, isto é, a construção se dá *de baixo para cima* a partir das folhas e segue em direção à raiz da árvore sintática.

Ambos os métodos costumam utilizar algum tipo de artifício para manter estados de retorno (*backup*) para as múltiplas opções de reescrita de uma regra. Assim, em caso de uma regra falhar na tentativa de construir um constituinte, é possível retornar para tentar utilizar uma outra alternativa.

Também é possível a construção de analisadores *top-down* e *bottom-up* mais eficientes, sem retornos (*backtracking*), dentre os quais citamos os analisadores *preditivos* e analisadores *shift-reduce*, respectivamente. De uma maneira geral, veremos que estes últimos são ditos determinísticos, porque não buscam alternativas na construção dos constituintes sintáticos. Existe uma extensa literatura que aborda estas técnicas, onde destacamos o trabalho de Aho, Sethi e Ullmann [1986].

Estes métodos apresentam vantagens e desvantagens. O método *top-down* é muito popular porque analisadores eficientes podem ser contruídos mais facilmente utilizando esta técnica. Porém, o analisador pode ficar muito tempo reescrevendo regras antes de considerar palavras da sentença, além de repetir diversas vezes uma mesma construção na busca da estrutura gramatical correta.

Por outro lado, os analisadores *bottom-up* podem tratar um número maior de gramáticas. Neste caso não existe repetição na análise de constituintes, mas o analisador considera todas as classes gramaticais das palavras e constrói estruturas que nunca levam a sentenças corretas.

Como uma forma de contornar os problemas de um e outro método, Allen [1987] sugere a construção de um analisador misto que reúne as vantagens de ambas as técnicas - um analisador *top-down* que não repete a construção de constituintes, utilizando uma tabela para armazenar os constituintes já construídos.

Preferências Semânticas

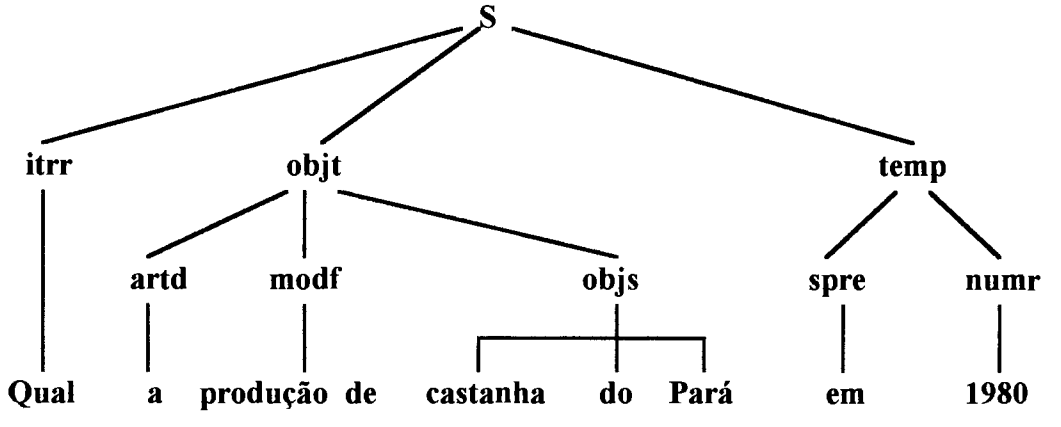
Os algoritmos de análise *top-down* e *bottom-up* dependem de técnicas de busca para encontrar possíveis estruturas gramaticais e interpretações para a sentença. Contudo, alguns estudos experimentais em psicolinguística apontam para o fato de que os processos humanos de análise e interpretação da linguagem são determinísticos, isto é, não procuram alternativas, mas ao invés disso usam a informação disponível no momento para construir a interpretação adequada.

Em particular, alguns resultados relativos a análise de sentenças com estruturas gramaticais ambíguas sugerem que, se as pessoas escolhem uma interpretação sem levar em conta as ambiguidades, então algumas interpretações devem ser preferidas em detrimento de outras. Assim, os estudos indicam a possível existência de certas "preferências semânticas" humanas no processamento da linguagem.

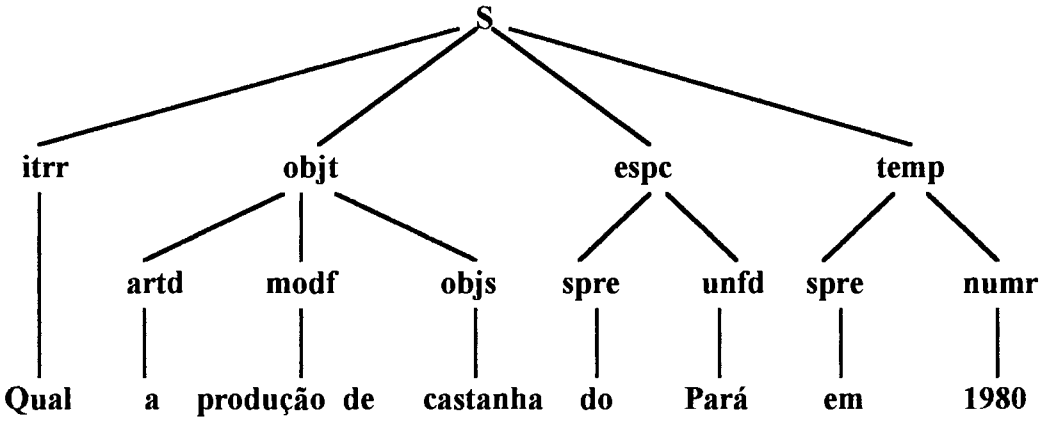
Dois tipos de preferências semânticas são especialmente notadas, e foram inicialmente sugeridas por Kimball [1973]. A primeira, conhecida como *ligação mínima*, é o princípio que envolve uma preferência por análises sintáticas que criam o menor número de nós na árvore sintática. O segundo tipo de preferência é conhecido como *associação a direita*, e se refere a tendência de que novos constituintes são interpretados como parte do constituinte que estiver sendo construído.

Para ilustrar, considere o enunciado "Qual a produção de castanha do Pará em 1980?" A seguir temos duas árvores sintáticas possíveis para a sentença:

Análise 1:



Análise 2:



Neste caso, ambos os princípios, de ligação mínima e associação a direita, favorecem a primeira análise, pois esta possui o menor número de nós na árvore sintática, além de interpretar a expressão "do Pará" como parte do constituinte *objs* formando o sintagma "castanha do Pará".

Porém, estas análises não se aplicam ao domínio semântico do banco de dados da pesquisa agrícola PAM: a "castanha do Pará" não é considerada na pesquisa, portanto não tem significação no contexto do banco de dados. Pelo fato da análise ser orientada à semântica, quando o analisador encontra a palavra "castanha", procura pelos complementos especificados no léxico (no campo *rpal*), que, neste caso, são as palavras "de cajú". Não encontrando tais complementos o sistema emite um erro, dizendo que não é possível continuar a análise da sentença após a palavra "castanha".

Além de ligação mínima e associação a direita, existem outros tipos de preferências semânticas, por exemplo quando o verbo decide a construção mais adequada. Cada verbo pode ser classificado segundo as preposições que introduzem os sintagmas preposicionais como complementos verbais, indicando que complementos podem estar associados ao sintagma verbal. Caso contrário, o sintagma preposicional deve ser considerado um complemento nominal, e associado a um sintagma nominal de acordo com o princípio de associação a direita. Esta estratégia é conhecida como *preferência léxica*. Considere as sentenças:

"Informe a produção de borracha no Pará."
"Que estados produzem castanha do Pará?"

No primeiro caso, o verbo admite um complemento de local, logo a expressão "no Pará" pode ser classificada, conforme indica a preferência léxica, como um adjunto adverbial de lugar que preenche o caso espacial da gramática. Contrariamente, o verbo da segunda sentença não admite um complemento de local, fazendo com que a única análise possível seja "do Pará" como complemento de "castanha", segundo o princípio de associação a direita.

Na interface AGRON, as preferências léxicas favorecem a estrutura de casos espaço-tempo. Daí a importância das preposições, pois estas marcam os complementos verbais válidos que preenchem a estrutura de casos.

Por outro lado, a formação dos sintagmas nominais está ligada à manipulação da entrada léxica associada ao primeiro substantivo de uma expressão, onde se encontra a informação de quantos (no campo *nexp*) e quais (no campo *rpal*) são os vocábulos válidos como complementos nominais do substantivo.

Analisadores Determinísticos

Visando aproximar o algoritmo de análise dos processos psicolinguísticos humanos, muito esforço tem sido dedicado no sentido de incorporar estes três princípios de preferências semânticas aos formalismos existentes, tais como Rede de Transição Aumentada e Gramática Livre de Contexto Aumentada. Além disso, diversas técnicas foram desenvolvidas para aumentar a eficiência dos algoritmos *top-down*, *bottom-up* e outros baseados em técnicas de busca.

Porém, mais importante para nós, é a ênfase no modelo determinístico de análise, onde os estados do analisador são usados para codificar temporariamente as ambiguidades até que uma porção maior da sentença seja analisada, adiando assim a decisão sobre a escolha das regras, e possibilitando que somente as regras que contribuem para a análise final sejam aplicadas. O mecanismo que permite ao analisador a inspeção de um número fixo de terminais antes de decidir pela regra é chamado de *olhar-adiante* (*lookahead*).

Existe uma classe de analisadores determinísticos implementados com uma técnica conhecida como *shift-reduce*, que costumam ser utilizados em gramáticas determinísticas como as das linguagens de programação. Estes analisadores procuram *reduzir* a sentença de entrada no símbolo inicial da gramática, utilizando uma construção *bottom-up* para a árvore sintática. A cada redução, uma cadeia de terminais e/ou não-terminais específica que casa com o lado direito de uma regra é substituída pelo lado esquerdo daquela regra, e, se o olhar-adiante é suficiente para escolher a cadeia correta, o processo continua sucessivamente até o símbolo inicial. São analisadores bastante eficientes e podem ser usados com sucesso para lidar com um grande número de Gramáticas Livre de Contexto. A eficiência da técnica reside no fato de que todas as possibilidades são consideradas com uma certa "antecedência", ditada pela extensão do olhar-adiante.

Por outro lado, temos os analisadores *preditivos*, onde dada uma sequência de entrada iniciada pelo terminal t, utiliza-se uma busca *top-down* a fim de decidir qual a alternativa das regras é a única que deriva a cadeia começada por t. A decisão, neste caso, é baseada na inspeção adiante de k símbolos terminais que o lado direito das regras pode derivar, ou seja, a seleção da regra é guiada pelo mecanismo de olhar-adiante. Este tipo de analisador é também muito eficiente, embora possa ser utilizado num conjunto menor de gramáticas em relação ao analisador *shift-reduce*.

Os analisadores *shift-reduce* e *preditivos* trabalham com classes de Gramáticas Livres de Contexto (GLC) que são expressivas o bastante para descrever a maioria das construções sintáticas das linguagens de programação. Embora a Linguagem Natural esteja longe de poder ser descrita por uma GLC e ser analisada por um analisador determinístico, existem técnicas que podem ser estendidas e/ou adaptadas para muitas das construções utilizadas nas interfaces baseadas em Linguagem Natural, como fazemos no caso da interface AGRON.

As subclasses de gramáticas processadas pelos analisadores *shift-reduce* e analisadores preditivos são chamadas, respectivamente, de gramáticas LR(k) - processam a entrada da esquerda para a direita construindo a derivação mais a direita (*left to right rightmost derivation*) - e LL(k) - processam a entrada da esquerda para a direita construindo a derivação mais a esquerda (*left-to-right leftmost derivation*) - onde k é o número de símbolos para o olhar-adiante. Neste caso, os analisadores são guiados por uma tabela (*table-driven*) onde as entradas relacionam as regras que devem ser usadas em função do estado do analisador. Os analisadores implementados manualmente geralmente usam uma gramática LL, ao passo que os analisadores para gramáticas LR costumam ser implementados utilizando-se ferramentas que constroem a tabela do analisador automaticamente.

Outro tipo de analisador diferente para Linguagem Natural, que não é baseado em buscas *top-down* ou *bottom-up*, é o PARSIFAL [Marcus, 1980], também conhecido como "analisador de Marcus", onde a operação de análise depende inteiramente do olhar-adiante. O analisador possui um conjunto de ações mais complexo do que as ações *shift-reduce* (ajustar-reduzir), construindo os constituintes através de um processo incremental, e não através de operações de redução na pilha de análise.

O interessante neste analisador é que, considerando as limitações impostas pelo mecanismo de olhar-adiante e pela natureza determinística do analisador, uma questão permanece para ser testada experimentalmente: "Será que as construções que o analisador não consegue processar corretamente são as mesmas que as pessoas têm dificuldade?" A questão se baseia no fato de que este método de análise trabalha com olhar-adiante em termos de constituintes, e não de palavras na entrada, o que torna o mecanismo muito mais poderoso, ampliando a capacidade de processamento de ambiguidades, o que, supostamente, o aproxima da capacidade do próprio ser humano.

O Analisador do AGRON

O método de análise usado no protótipo da interface AGRON é chamado "análise recursiva descendente" (*recursive descent parsing*) [Aho, Sethi & Ullmann, 1986]. É uma técnica *top-down* onde executa-se um conjunto de procedimentos recursivos para processar a sentença de entrada. Neste caso, um procedimento é associado com cada não-terminal da gramática. Além disso, o método é determinístico porque utiliza um conjunto $k <= 2$ de símbolos terminais para olhar-adiante, ao decidir de forma não-ambígua o procedimento que deve ser executado a partir de cada não-terminal. Desta forma, cada constituinte é construído uma só vez, e somente aqueles que fazem parte da análise correta são apresentados como resultado.

Na verdade, este é também um analisador preditivo, conforme descrito na seção anterior, só que o controle é implícito, através de chamadas recursivas de procedimentos, ao invés da manutenção explícita de uma tabela, como ocorre nos analisadores guiados por tabela (*table-driven*).

Embora a gramática permita a utilização de qualquer dos tipos de analisadores descritos anteriormente, a escolha da técnica de análise recursiva descendente deve-se, por um lado, à simplicidade de especificação do algoritmo numa linguagem procedimental como o PASCAL, e por outro, à perfeita adequação a uma gramática semântica de casos. No primeiro nível de análise, os procedimentos corresponderão a estrutura de casos semânticos, e no segundo nível de análise, estarão ligados às ações semânticas que mapeiam os não-terminais em estruturas do banco de dados.

O olhar-adiante até dois terminais ($k = 2$) pode ser identificado nas indentações de *if*s do algoritmo, onde ocorrem chamadas sucessivas do *Scan* em até dois níveis, testando os não-terminais (associados à entrada léxica do terminal através do campo *nterm*) antes de decidir o procedimento adequado.

Dispensaremos a especificação dos testes para verificação de erro porque sobrecarregariam o trecho do algoritmo que reproduzimos aqui. Existem dois tipos de erros tratados pelo analisador: erros provenientes do *Scan*, tal como não encontrar a palavra no léxico, e erros de análise estrutural, onde ocorre um constituinte numa posição não esperada. Segue-se o trecho do algoritmo de análise correpondente ao procedimento **ParseFrase**, responsável pela análise do primeiro nível da sentença, que identifica os casos *objeto*, *espaço*, *tempo* e *restrição*:

```
Procedimento ParseFrase

Enquanto não é fim de frase e não tem erro de análise
faça
    Caso o não-terminal associado à entrada léxica seja
    smun, sunf, sreg, sano, sobj: ParseObjeto
     { substantivo  sempre inicia um caso objeto}
    detm:  Scan
           ParseObjeto
     {determinador sempre inicia um caso objeto }
    cond:  Scan
           ParseRestrição
     {cond (pron. relativo) sempre inicia um caso restrição}
    cpre: ParseTempo
     { preposições cpre sempre iniciam um caso temporal }
    ppre:  Scan
           Se não-terminal é detm
           então
                  Scan
                  Se não-terminal é sano
                  então ParseTempo
                  senão
                         Se não-terminal é smun, sunf, sreg,
                         munc, unfd, regi ou bras
                         então ParseEspaço
                         senão MsgErro
           senão
                  Se não-terminal é numr
                  então ParseTempo
                  senão
                         Se não-terminal é munc, unfd ou regi
                         então ParseEspaço
                         senão MsgErro
    spre:  Scan
           Se não-terminal é sano ou numr
           então ParseTempo
           senão
                  Se não-terminal é smun, sunf, sreg,
                  munc, unfd, regi, bras
                  então ParseEspaço
                  senão MsgErro
    senão: MsgErro

Se não tem erro
então Escreva: "Sentença foi analisada com sucesso"
fim
```

X. ANÁLISE SEMÂNTICA

Segundo Allen [1987], é desejável ter-se uma representação semântica intermediária entre a forma sintática da sentença e a forma final produzida pela interface em LN, porque tal representação estabelece uma divisão entre dois problemas separados, mas não independentes, a saber:

- a escolha entre os múltiplos sentidos das palavras numa sentença de maneira que a sentença como um todo adquira o significado pretendido;
- o uso do conhecimento do mundo e do contexto para identificar e modelar as consequências e efeitos de uma tal sentença.

O primeiro problema, conhecido como "disambiguação léxica" é justamente o processo de eliminar as ambiguidades no que concerne às acepções inadequadas das palavras no enunciado corrente, a fim de gerar a referida interpretação intermediária, comumente chamada de *forma lógica*. O segundo problema está relacionado com as aplicações onde algum tipo de interpretação contextual da sentença é necessária, a fim de modelar possíveis inferências e implicações mais complexas, como ocorre no processamento de discursos.

Como mencionamos inicialmente, o uso de interfaces em LN para acesso a banco de dados em geral não requer um tratamento contextual sofisticado das sentenças, já que este é simplificado pelo próprio uso do "mundo fechado do banco de dados. Porém, mesmo as interfaces *domain-dependent*, e sobretudo as interfaces *transportáveis*, mantém uma representação intermediária com a finalidade de garantir a separabilidade entre as fases *front end* e *back end* do sistema, ou seja, entre as fases que dependem principalmente da sentença de entrada e são independentes da aplicação, e as fases que dependem da aplicação, do banco de dados, e/ou da linguagem de *query* alvo. Neste caso, o *front end* normalmente corresponde as fases de análise léxica, sintática e parte da análise semântica, enquanto o *back end* equivale a porção complementar da interpretação semântica mais a fase de formalização. Desta forma, é conveniente termos uma representação intermediária produzida pelo *front end* da interface que possa ser utilizada por diversas configurações de *back end*, estas últimas podendo ser entendidas como o mapeamento para um domínio diferente, ou para uma linguagem de *query* alternativa.

Na verdade, a atual implementação interface AGRON não adota uma forma lógica especial, mas efetua uma tradução direta para uma forma simplificada da linguagem SQL (Structured Query Language). Contudo, podemos dizer que a forma do SQL utilizada funciona como uma representação intermediária que atende perfeitamente aos requisitos de portabilidade para outros domínios com estrutura espaço-temporal. O uso de uma forma lógica é uma extensão que propomos mais adiante.

Representação do Conhecimento

O problema de gerar uma representação semântica intermediária é um processo complexo que não pode ser resolvido simplesmente através de um método dedutivo, usando apenas o cálculo de predicados. Como ressalta Allen [1987], "os componentes de significação - significado de palavras, frases, entre outros - geralmente correspondem, no máximo, à fragmentos de uma lógica, tais como nomes de predicado, tipos, e outros sobre os quais a dedução não está definida". Para ilustrar, voltemos às sentenças:

1: "João quebrou a janela com o martelo."
2: "O martelo quebrou a janela."
3: "A janela quebrou."

Naturalmente, poderíamos usar o predicado *quebrar*(x,y,z) para modelar o significado do verbo, onde x é a pessoa que quebra, y é a coisa quebrada, e z o instrumento utilizado para quebrar. Contudo, para mapear os argumentos do predicado é necessário utilizar informação sintática e semântica sobre os sintagmas nominais "João", "a janela" e "o martelo" para decidir que o sujeito corresponde ao primeiro argumento x na sentença 1, ao terceiro argumento z na sentença 2, e ao segundo argumento y na sentença 3. Conforme podemos observar, dificilmente será possível descrever este procedimento usando um processo dedutivo.

Logo, são necessárias formas adicionais de representação do conhecimento a fim de codificar a semântica de objetos e ações no mundo. Algumas formas conhecidas de representação do conhecimento são *redes semânticas*, *frames* e *scripts*, onde a primeira nos interessa particularmente.

Redes Semânticas e Representação de Taxonomias

Uma das teorias mais conhecidas para representação de semântica são as chamadas Redes Semânticas (RS), usadas para os mais diferentes propósitos, dando origem assim a diversos formalismos.

De uma forma geral, uma Rede Semântica é definida como um grafo com nós e elos rotulados que designam, respectivamente, entidades e relações as quais se deseja representar. Além disso, alguns sistemas associam uma semântica formal aos componentes da RS a fim de mapear a estrutura numa fórmula equivalente ou aproximada do cálculo de predicados. Muitos formalismos de RS oferecem uma capacidade restrita de representar fórmulas quantificadas, conjunções e disjunções. Contudo, muitos autores têm desenvolvido formalismos mais sofisticados, com um poder representativo superior ao do Cálculo de Predicados de Primeira Ordem. Entre eles estão Hayes [1977], Hendrix [1979], Brachman [1979] e Woods [1991].

Muitos tipos de RS refletem o conhecimento sobre *classificações* onde os objetos são descritos numa hierarquia e agrupados de acordo com certas propriedades. Além disso, existem relações de natureza diversa que dão origem as variadas espécies de elos.

Uma propriedade muito difundida se relaciona ao chamado *tipo* do objeto, e a taxonomia que considera este *tipo* é conhecida como *hierarquia de tipos*. A hierarquia de tipos mais conhecida é a que descreve os objetos do mundo físico dividindo-os em coisas animadas e inanimadas, as coisas animadas em animais e vegetais, e assim por diante. Porém, outras entidades podem corresponder a tipos. Como vimos, os tipos de verbos, podem ser organizados em hierarquias que variam com o aspecto semântico que retratam. Além disso, muitas vezes os tipos não podem ser organizados em hierarquias simples, mas em redes mais sofisticadas, porque guardam entre si relações mais complexas de interseção, exclusão mútua, e outras.

No caso particular do AGRON, temos a hierarquia associada às subdivisões do espaço territorial brasileiro e a hierarquia de produtos agrícolas. Esta última, foi enquadrada numa hierarquia mais complexa segundo o formalismo de subsunções. A vantagem da taxonomia de subsunções em relação à hierarquia de tipos é que na primeira existe uma definição de *conceito* que consideramos mais pertinente, ao passo que na última persiste uma questão, discutida por muitos pesquisadores, do que realmente um *tipo* pode representar. Além disso, como já dissemos anteriormente, na hierarquia de subsunções existe uma distinção clara entre os elos definicionais ou estruturais e os elos assertivos, e uma série de operadores que aumentam o poder semântico do formalismo. Poderíamos, inclusive, utilizar somente a taxonomia de subsunções de Woods para representar ambas as hierarquias, pois entendemos que esta abrange a capacidade representativa da hierarquia de tipos convencional.

Na implementação vigente da interface AGRON, a semântica que estaria representada numa rede semântica de subsunções está, na verdade, distribuída nos diversos níveis de análise. Porém, a especificação de uma rede semântica para o AGRON permanece como uma proposta viável e muito interessante de extensão do sistema.

A Análise Semântica no AGRON

A técnica de análise semântica usada para interpretar LN no AGRON permite informação sintática e semântica codificada nas regras da gramática, no léxico, e na maioria dos procedimentos do analisador. Assim sendo, a análise semântica começa a ser aplicada diretamente na sentença de entrada com o mapeamento para as classes semânticas, e posteriormente na integração destas em estruturas que vão compor os casos *objeto*, *espaço*, *tempo* e *restrição*. Além disso, algumas regras semânticas específicas para o domínio da aplicação são usadas para mapear verbos e condições para os comandos correspondentes na representação final em SQL. Neste caso, dizemos que o analisador é "guiado pela semântica" (*semantically-driven*).

Em particular, o léxico, conforme vimos, contém informações semânticas essenciais, tais como os diversos sentidos de uma palavra, as classes semânticas e os adjetivos associados. Além disso, o analisador léxico possui uma rotina para disambiguação das palavras e uma rotina que verifica o agrupamento das palavras em expressões com uma significação especial no domínio em questão.

Na verdade, a característica principal da interface é a orientação através de informação semântica em todos os níveis de análise, e por isso, a avaliação de grande parte dos aspectos semânticos já foi fornecida previamente nas seções sobre processamento de verbos, estrutura de casos e na descrição de cada caso em particular, além dos mecanismos descritos nos capítulos sobre a gramática, análise léxica e análise sintática. Resta-nos ainda, contudo, a descrição de certas regras semânticas especiais associadas com a tradução para SQL, que serão discutidas no próximo capítulo.

Ambiguidades

O tratamento de ambiguidades é um aspecto muito enfatizado nas pesquisas sobre LN. Neste caso, o problema maior reside nos métodos, geralmente não-determinísticos, usados pelos sistemas para eliminar os sentidos ambíguos de palavras e sentenças. Vale lembrar que, enquanto a desambiguação é um problema difícil no processamento artificial da linguagem, os seres humanos simplesmente parecem não considerar estas ambiguidades no processo de compreensão da língua.

As ambiguidades podem ser divididas em dois tipos fundamentais: léxicas e estruturais.

A ambiguidade léxica ocorre quando é preciso decidir entre os diversos sentidos de uma palavra para que a frase como um todo tenha o significado pretendido no contexto. Como dissemos, este problema é conhecido como *desambiguação léxica*, e constitui um dos procedimentos principais na fase de análise semântica das sentenças.

No sistema AGRON, a ambiguidade léxica ocorre em duas situações.

- Situação 1: A palavra possui, isoladamente, mais de um sentido, acarretando a presença de entradas léxicas com classes gramaticais e semânticas diferentes para as diversas acepções da palavra. Esta é uma ambiguidade léxica convencional. Exemplos:

a	artigo definido	detm
a	preposição	dpre
que	pronome interrog.	itrr
que	pronome relativo	cond
onde	advérbio interrog.	itrr
onde	pronome relativo	cond

- Situação 2: A palavra não possui múltiplos sentidos, mas em conjunto com outras palavras, forma expressões com significados específicos no contexto. O dicionário possui então tantas entradas léxicas quantas são as expressões onde a palavra participa, sendo que as classes gramatical e semântica podem ser as mesmas em todas as entradas, porque o que define a entrada léxica é a expressão inteira. Embora a incidência desta situação seja maior do que a primeira, esta não é uma ambiguidade léxica no sentido usual, porém trata-se também de decidir qual é a entrada léxica correta, com base no *olhar-adiante* das palavras que compõem a expressão. Exemplos:

algodão arbóreo	sintagma nominal	objs
algodão herbáceo	sintagma nominal	objs
área coletada	sintagma nominal	modf
área plantada	sintagma nominal	modf
arroz	substantivo	objs
arroz irrigado	sintagma nominal	objs
arroz sequeiro	sintagma nominal	objs
batata doce	sintagma nominal	objs
batata inglesa	sintagma nominal	objs
cana forrageira	sintagma nominal	objs
cana-de-açúcar	sintagma nominal	objs
lavoura permanente	sintagma nominal	objc
lavoura temporária	sintagma nominal	objc

A ambiguidade estrutural ocorre quando existe mais de uma forma de agrupar os constituintes de uma sentença. Temos então, como descrevemos previamente, uma ambiguidade estrutural na sentença:

"Qual a produção de castanha do Pará em 1980?"

onde o sintagma "do Pará" pode ser um complemento nominal determinando o tipo de castanha, ou um adjunto adverbial de lugar designando o local onde se deseja conhecer a produção.

As ambiguidades também costumam ser classificadas em sintáticas, semânticas ou contextuais, segundo o tipo de conhecimento usado para resolvê-las. Assim, a sentença ambígua do último exemplo caracteriza uma ambiguidade semântica porque é a preferência semântica ditada pelo verbo que é utilizada para decidir o significado pretendido da frase.

Também podem ser entendidas como ambiguidades semânticas as palavras contidas nas diversas expressões do quadro anterior da *situação 2*, que caracterizam objetos e seus atributos no domínio de aplicação do AGRON. Neste caso, é o papel semântico da expressão que importa na decisão sobre a entrada léxica. Este tipo de conflito é resolvido na rotina *ExpSemant* do analisador léxico.

As ambiguidades sintáticas podem ser encontradas nas palavras do dicionário fixo com múltiplas classes gramaticais, apresentadas no quadro da *situação 1*. Neste caso, é a classe gramatical que é usada para desambiguação, e a rotina léxica responsável pelo procedimento de resolução é *Ambiguidade*.

Em particular, as ambiguidades contextuais da linguagem do AGRON podem ocorrer em enunciados elípticos tais como:

"E a produção?"

onde é necessário um conhecimento da situação para entender que nos referimos a produção de algum produto agrícola, certamente referido num enunciado anterior, e não a uma produção de cinema, por exemplo.

A resolução de ambiguidade no sistema AGRON não é um processo árduo devido ao uso de uma gramática semântica. Contudo, outras aplicações de LN exigem métodos mais sofisticados de resolução de ambiguidades léxicas, e por isso o problema continua sendo objeto de diversas pesquisas na área de processamento de Linguagem Natural.

Gostaríamos de citar como uma proposta interessante o recente trabalho sobre desambiguação léxica usando redes Bayesianas descrito na dissertação de Alves [1993]. Neste sistema, a resolução de ambiguidades é fruto de um paralelismo entre as análises sintática e semântica, e da forma como são representados os diversos sentidos da palavra na rede Bayesiana. A representação semântica é um misto dos formalismos da dependência conceitual de Schank e da taxonomia de subsunções de Woods. Os nós na rede correspondem a variáveis do modelo probabilístico, cuja ocorrência corresponde a um padrão sintático ou semântico. A simulação da rede determina qual o padrão global frequente, indicando quais os significados adequados das palavras na frase. Além disso, o fato da simulação ser estocástica permite o contorno do problema da complexidade do cálculo da configuração mais provável.

Escopo de Quantificadores

Muitas ILNs tratam do problema da determinação do escopo dos quantificadores, embora esta não seja a ênfase na atual implentação da interface AGRON. A determinação do escopo de quantificadores associados a uma forma lógica da sentença é fundamental na obtenção da resposta correta. Por exemplo, considere a seguinte sentença:

"Todo homem ama uma mulher."

Neste caso, temos duas formas lógicas possíveis cujas paráfrases são:

- "para todo h tal que h é homem, então existe m tal que m é mulher e h ama m"
- "existe m tal que m é mulher e para todo h, h é homem e h ama m"

Na primeira, existe uma mulher para cada homem e o homem ama esta mulher; na segunda, existe uma única mulher que todo homem ama. São duas interpretações completamente diferentes que fornecem informações inteiramente distintas.

Existem algoritmos baseados em heurísticas que tentam determinar corretamente o escopo dos quantificadores na forma lógica. Por exemplo, a determinação do escopo dos quantificadores na interface TEAM segue a mesma estratégia utilizada no sistema LADDER [Hendrix, 1978] com algumas modificações. O algoritmo gera todas as possibilidades de escopo (baseado na estrutura sintática da sentença) e então testa cada alternativa submetendo-a a um processo chamado de "críticas de escopo". Cada crítica atribui uma pontuação a cada alternativa, e a alternativa que possuir a melhor pontuação resultante de todas as críticas é a escolhida.

A cada quantificador da forma lógica é associado um valor representando a *força* do quantificador. As críticas obedecem o seguinte critério: se não existe nenhuma instância para a forma lógica do escopo em questão, então uma pontuação 0 é atribuída; valores positivos aumentam a chance de um escopo e valores negativos diminuem a chance. São cinco tipos de crítica, que consideram desde a força do quantificador até problemas específicos do quantificador associado a determinadores do tipo "qualquer".

Os mecanismos para determinação do escopo de quantificadores dos sistemas LUNAR e CHAT-80 fazem parte das regras de construção da forma lógica. A diferença entre os algoritmos do TEAM e dos algoritmos do LUNAR e do CHAT-80 é que o TEAM gera e compara todos os escopos possíveis, enquanto o LUNAR e o CHAT-80 aplicam repetidamente algumas regras de escopo a uma representação intermediária a fim de produzir uma forma lógica final que satisfaça todas as regras de escopo relevantes. Ou seja, LUNAR e CHAT-80 usam algoritmos *first fit*, enquanto o TEAM usa um algoritmo *best fit*.

Embora um conjunto de regras tente determinar corretamente o escopo dos quantificadores da sentença, algumas vezes, certas considerações pragmáticas fornecem melhores resultados que a aplicação das regras. Por exemplo, na pergunta:

"Qual é a produção agrícola de todos os estados do Brasil?"

o sistema poderia interpretar que se procura por uma produção que seja de todos os estados ao mesmo tempo. Esta seria uma interpretação válida para sentenças do tipo "Quem é o chefe de todas as tropas?", porém no caso anterior é a pragmática que nos diz que cada estado tem a sua própria produção.

XI. FORMALIZAÇÃO

O processo de formalização no protótipo da interface AGRON consiste na tradução da estrutura de casos semânticos num comando da linguagem SQL que possa ser submetido para *query* no banco de dados da produção agrícola (BMA). Neste caso, o procedimento de tradução ainda está relativamente ligado ao domínio do banco de dados, embora o uso de um dicionário poderoso e de uma gramática semântica orientada por uma estrutura de casos permitam a flexibilidade de adaptação a outros domínios espaço-temporais que se assemelhem ao padrão definido para o atual domínio do AGRON.

Antes de descrever as regras de tradução é necessário fornecer uma visão do banco de dados utilizado para que possamos compreender como se opera o mapeamento das estruturas fornecidas pelo analisador nos componentes da linguagem de *query*. Na verdade, o banco de dados pode ser apresentado em duas visões distintas porém relacionadas:

- o nível conceitual, mais genérico, onde os objetos são definidos em termos de entidades e relacionamentos, sem a preocupação com o SGBD onde o banco será implementado;
- o nível físico, contendo a descrição das relações dos arquivos físicos, que resulta então de operações de normalização e adaptação das estruturas definidas no nível anterior aos requisitos do SGBD.

Desta forma, podemos ter a especificação de dois esquemas, o conceitual e o físico, dependendo do aspecto que precisamos enfocar. Além disso, observamos que, no caso da interface AGRON, as entidades extensionais estão descritas nas relações que compõem o banco de dados, ao passo que as entidades intensionais estão descritas no léxico e nas regras de reescrita da gramática semântica.

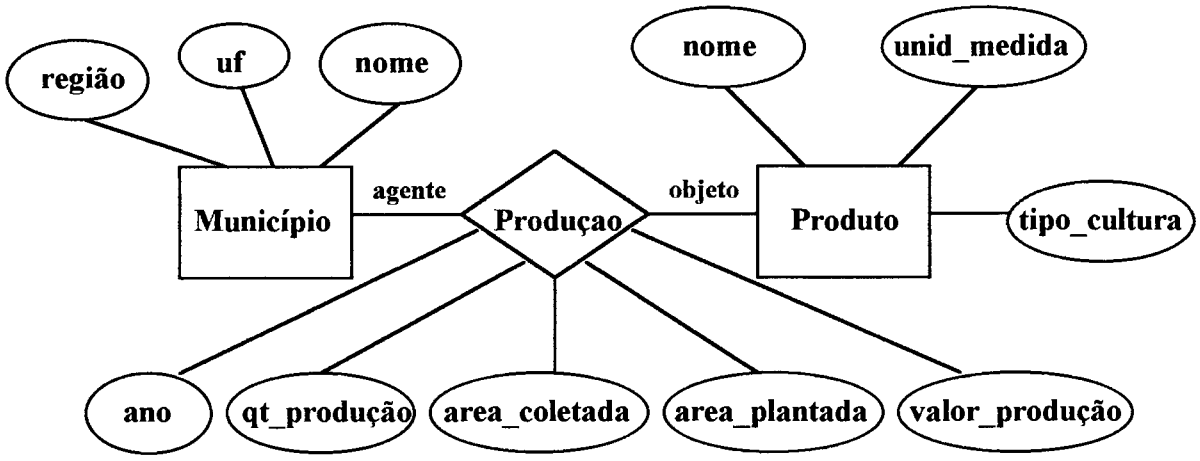
O Modelo de Entidades e Relacionamentos

O chamado Modelo de Entidades-e-Relacionamentos (MER) é comumente usado para representar o esquema conceitual do banco de dados. O Modelo de Entidades-e-Relacionamentos considera o mundo como composto de *entidades* e *relacionamentos* entre estas entidades. Neste caso, uma *entidade* é qualquer coisa identificada unicamente que tenha relevância no contexto, e um *relacionamento* é uma associação entre as entidades. Assim, existem tipos de entidades, tipos de relacionamentos e o conjunto de atributos associados a um esquema conceitual em particular, sendo que o conjunto de atributos, descreve o domínio das entidades e dos relacionamentos.

Num diagrama MER, um tipo de entidade é representado por um retângulo contendo o nome da entidade, e um tipo de relacionamento é representado por um losango com o nome do relacionamento. As entidades que participam num relacionamento são indicadas através de linhas de ligação que podem ser rotuladas com informações semânticas sobre o relacionamento, tais como os casos ou papéis semânticos. Os conjuntos de atributos são representados em elipses contendo o nome do atributo associado.

O modelo conceitual do BMA é composto de duas entidades básicas, o *município* e o *produto*, mais o relacionamento de *produção agrícola* entre estas. A seguir temos o MER do Banco de Microdados Agropecuários (BMA) correspondente a modelagem da produção agrícola. Omitiremos a porção da modelagem que descreve a produção pecuária, pois o protótipo do AGRON visa contemplar primeiramente a produção agrícola. Contudo, a modelagem da produção pecuária é muito semelhante, pois as entidades envolvidas são as mesmas, alterando-se apenas os atributos das entidades e do relacionamento. Consequentemente, a extensão do AGRON para este último domínio, o da pecuária, é realmente muito simples, dependendo basicamente de um aumento do léxico.

Modelo de Entidades e Relacionamentos do BD da Produção Agrícola



Relações do Banco de Dados

Existe uma tendência na área de banco de dados que aponta o modelo relacional como o mais adequado para armazenar dados estruturados em virtude da flexibilidade oferecida pelas operações sobre as relações, possibilitando um processamento eficiente através de um SGBD poderoso como o DB2, por exemplo. Além disso, o uso de tabelas para representar relações é uma forma direta e transparente de codificar os dados, pois esta parece ser a forma como os usuários naturalmente visualizam as informações.

O fato é que uma boa parte das ILNs, transportáveis ou *domain-dependents*, pré-supõe o uso subjacente de uma estrutura relacional de banco de dados. Este fato, simplifica, por exemplo, o problema da aquisição de verbos. Neste caso, os argumentos dos predicados verbais são sempre valores de algum atributo de alguma relação.

No modelo relacional de banco de dados existem os conceitos de *relação de entidades* e *relação de relacionamentos* para denominar as tabelas que organizam, respectivamente, os atributos de um tipo de entidade e os atributos de um tipo de relacionamento.

No BMA, existem duas relações, a saber, a relação *agrícola*, da produção agrícola propriamente dita, resultante do relacionamento entre produto e município, e a relação *prodpam*, de produtos agrícolas da PAM. As informações específicas da entidade município não integram diretamente o banco de dados, mas os códigos de município e unidade da federação estão contidos na relação *agrícola*, donde também é possível inferir a região a qual o município pertence. Desta forma, um subconjunto das relações do BMA pode ser visualizado nas tabelas:

uf	município	produto	ano	produção	área_coletada	área_plantada	valor_produção
15	Belém	0150	79	150000	20000	22000	34000000,00
15	Belém	0150	80	220000	35000	38000	50000000.00
15	Belém	0280	79	20	15000	15000	22500000,00
15	Belém	0280	80	24	15000	16000	32000000,00
32	Vitória	0150	79	90000	10000	12000	18000000,00
32	Vitória	0150	80	120000	13000	15000	33000000,00
32	Vitória	0720	80	30	17000	20000	27000000,00
35	São Paulo	0150	79	125000	18000	20000	24000000,00
35	São Paulo	0720	79	15	9000	15000	9500000,00
35	São Paulo	0720	80	25	18000	20000	15000000,00

Relação *Agrícola*

produto	nome_produto	unid_medida	tipo_cultura
0150	banana	cachos	temporária
0280	castanha-de-cajú	toneladas	permanente
0720	tomate	toneladas	temporária

Relação *ProdPAM*

Tradução para o SQL

A linguagem SQL (Structured Query Language) é usada para acessar bancos de dados relacionais. Diferentemente de outras linguagens procedimentais de *query*, onde é necessário codificar de uma sequência de instruções, o acesso através da linguagem SQL é feito através de um único comando dirigido ao SGBD.

Contudo, temos o argumento de Ott [1991] no seu artigo "Aspects of the Automatic Generation of SQL Statements in a Natural Language Query Interface" de que, para alcançar o poder da LN, muitas vezes é necessário mapear a *query* em LN em múltiplas expressões SQL, tantas quantas forem necessárias para responder a questão. Ele sugere estratégias que são especialmente úteis se ocorrem, simultaneamente, quantificadores e negação numa sentença. Além disso, afirma que é estritamente necessário um aumento do SQL, que ele denomina SQL +, para possibilitar o processamento de sentenças com ordinais, como em "o segundo maior estado produtor", porque até agora não existe um operador SQL que permita lidar com tais construtos.

Em seu artigo, Ott descreve o módulo de geração de SQL do sistema *LanguageAcess*, um programa licenciado pela IBM e disponível atualmente para o inglês e o alemão. Este sistema usa, de fato, duas representações semânticas intermediárias: CLF (*Conceptual Logical Form*) e DBLF (*DB Logical Form*). Contudo, Ott concentra-se na última forma lógica, DBLF, pois esta serve de entrada para o processo de geração automática de SQL. A DBLF assemelha-se ao cálculo de predicados de primeira ordem, aumentada com alguns operadores especiais e funções que são necessárias num ambiente de banco de dados.

Partindo da sentença representada em DBLF, Ott sugere três estratégias principais - a estratégia da junção (*join strategy*), a estratégia da relação temporária (*temporary relation strategy*) e a estratégia da negação (*negation strategy*) - além de uma mistura destas três. O objetivo é mostrar que estas estratégias viabilizam o mapeamento da sentença em LN numa sequência de comandos da representação aumentada do SQL, o SQL +, onde o tratamento de quantificadores numéricos e universais, além de outros construtos, pode ser feito de maneira sistemática e uniforme. A desvantagem, neste caso, é que nem sempre as expressões geradas no SQL + constituem a forma ótima, do ponto de vista do desempenho, de processar a *query*.

Os aspectos abordados por Ott são extremamente úteis quando considerados como estratégias gerais num ambiente de tradução para o SQL + a partir de uma forma lógica intermediária, porém têm pouca importância numa interface interativa de tradução direta para o SQL, como é o caso da interface AGRON.

No protótipo da interface AGRON, a tradução se dá automaticamente para a linguagem SQL. A sintaxe utilizada é uma simplificação do SQL voltado para processamento de *queries* simples, sem *subselect* ou *union*, resumida no seguinte comando:

SELECT *colunas* FROM *tabelas* WHERE *expressões*

onde *colunas* referenciam atributos das relações, *tabelas* descrevem o nome das relações, e *expressões* são condições de busca.

As regras de tradução, na forma de fragmentos de programa, são aplicadas ao longo dos procedimentos de análise da sentença, e geram as cláusulas da linguagem a partir dos valores para os casos semânticos obtidos no léxico que estejam disponíveis em um dado momento. O mapeamento para as cláusulas SQL obedece um esquema de tradução conforme descrito nas seguintes regras:

- Toda referência a um *objeto* e seus modificadores mapeia os correspondentes nomes de atributo nas *colunas* da cláusula *select*; se o objeto é uma referência definida, ocorre também, na cláusula *where*, uma condição adicional para selecionar a referida instância do objeto.

- Para os objetos produto agrícola, município, estado, região e ano, o mapeamento na cláusula *select* do comando SQL é para as colunas *agricola.produto*, *agricola.municipio*, *agricola.uf* e *agricola.ano*, respectivamente. Além disso, se ocorrem modificadores, tal como "área colhida", mapeia-se na cláusula *select* o nome da coluna associada, neste caso *agricola.area_coletada*.

- A região é deduzida a partir da coluna *agricola.uf*, de acordo com um intervalo de valores específico; por exemplo, a região sudeste compreende as ufs com códigos maiores que trinta e menores ou iguais a quarenta. Portanto, se o objeto é uma região, a coluna mapeada também é a coluna *agricola.uf*, sendo que, na cláusula *where*, acrescentamos a condição que descreve o intervalo para a referida região.

- Quando um objeto composto é referenciado, as expressões na cláusula *where* incluem a condição de busca para o tal objeto composto. Por exemplo, se a questão envolve os produtos da lavoura temporária, então ocorre a condição *prodpam.tipo_cultura = 't'* na cláusula *where*.

- Todas as referências definidas nos casos espaço, tempo e restrição mapeiam expressões na forma de condições de busca na cláusula *where*, e paralelamente, mapeiam um nome de atributo na coluna correspondente da

cláusula *select*.

- A tradução para as condições de busca dos casos tempo e restrição é feita diretamente a partir das entradas léxicas, ao passo que a tradução do caso espacial depende do processamento da expressão que denota o espaço territorial de acordo com a hierarquia de tipos apresentada anteriormente.

- O valor para o caso semântico *objeto* só admite um tipo de objeto sendo processado de cada vez. Além disso, verifica-se a compatibilidade entre o objeto e os casos espaço-tempo-restrição, isto é, se o objeto é um espaço territorial não pode haver um construto associado ao caso espacial na sentença.

- Toda coluna está associada a uma tabela, portanto as tabelas da cláusula *from* resultam do somatório das tabelas associadas às colunas referidas na cláusula *select*.

- As operações de *join* das tabelas são indicadas em estruturas fora do léxico, através de condições do tipo *agricola.produto = prodpam.produto* que devem ser acrescentadas à cláusula *where*. As condições são calculadas de acordo com regras de integridade definidas para as tabelas envolvidas na *query*, neste caso *agricola* e *prodpam*, e as colunas comuns, que neste caso se resume a coluna de *produto*.

Para ilustrar, considere a sentença em LN processada pela interface AGRON, e a respectiva tradução para o SQL, usando o MER (modelo de entidades e relacionamentos) descrito anteriormente para o BMA.

"Forneça a área plantada e a área colhida de uva para os estados da região sul onde o valor da produção foi superior a Cr$ 200.000,00 e inferior a Cr$250.000,00 em 1980."

Comando SQL:

```
Select agricola.produto   agricola.area_cultivada
agricola.area_coletada   agricola.sigla_uf
agricola.valor_producao  agricola.ano
from  agricola
where (agricola.produto = 0760) and
      (agricola.uf  > 40 and agricola.uf < 50) and
      (agricola.valor_producao > 200000 and agricola.valor_producao <
250000) and
      (agricola.ano = 80)
```

Do ponto de vista semântico, se observarmos a estrutura objeto-espaço-tempo-restrição dos casos da gramática do AGRON, podemos entender a tradução para o SQL como um mapeamento para a seguinte forma:

SELECT *objeto e modificadores* FROM *tabelas* WHERE *restrições e condições para espaço, tempo e objeto*

Considere o exemplo:

"Forneça a área plantada, área colhida e a produção de soja entre 85 e 91 em Goiás e na região Sudeste".

Esta sentença não pode ser processada pelo AGRON. Não serão aceitas conjunções e disjunções de espaços de níveis territoriais diferentes, tal como a referência a um estado e a uma região simultaneamente como no exemplo anterior "em Goiás e na região Sudeste".

Contudo, no presente escopo da interface AGRON, o uso de enunciados elípticos compensa estas limitações do sistema. Neste caso, ao invés do usuário entrar, por exemplo, com a especificação "em Goiás e na região Sudeste", ele pode elaborar o seu pedido usando sentenças distintas em iterações consecutivas com o sistema, sendo que a primeira seria uma sentença completa, e as seguintes, formas elípticas em relação a primeira. Por exemplo:

"Forneça a área plantada, a área colhida e a produção de soja entre 85 e 91 em Goiás"

e na iteração seguinte:
"E na região Sudeste?"

A seguir, temos o exemplo do algoritmo correspondente ao trecho de programa que define o procedimento MontaRest, responsável pela montagem e tradução das expressões relacionais do caso restrição.

```
{ Monta Restricoes }

  Procedure MontaRest;

  Enquanto não-terminal é um operador relacional  (nterm = qpre, mpre) ou
                não-terminal é um numeral       (nterm = numr) ou
                não-terminal é a conjunção 'e'  (nterm = cjun)  ou
                não-terminal é a disjunção 'ou' (nterm = djun)
  faça
    Se não-terminal = numeral
    então restricao := restricao + numeral + ' '
    senão
      Se palavra = 'igual'
      então restricao := restricao + '= '
      senão
        Se palavra = 'diferente'
        então restricao := restricao + '<> '
        senão
          Se (palavra = 'maior') ou
             (palavra = 'mais')  ou
             (palavra = 'superior')
          então restricao := restricao + '> '
          senão
            Se (palavra = 'menor') ou
               (palavra = 'menos')  ou
               (palavra = 'inferior')
            então restricao := restricao + '< '
            senão
              Se palavra = 'e'
              então restricao := restricao + 'and '
              senão
                Se (palavra = 'ou')
                então restricao := restricao + 'or '
                senão continua;
   Scan;
  fimfaça;

  Se não-terminal é unidade de medida  (nterm = unmd)
 { le unidade de medida }
  então Scan;

fim;
```

Tradução para a Forma Lógica

Algumas ILNs, assim como o protótipo da interface AGRON, transformam a sentença em LN diretamente num comando de uma linguagem de *query* para acesso ao banco de dados. A principal vantagem, neste caso, assenta-se no alto desempenho da interface.

Contudo, existe uma forte tendência em LN que aponta um esforço concentrado no desenvolvimento de sistemas mais versáteis, facilmente adaptáveis a novos domínios. Neste caso, como já foi dito, é recomendável uma representação semântica intermediária, geralmente baseada numa lógica formal, não somente para adquirir portabilidade, mas também devido a modularidade que tal representação proporciona, permitindo que o mapeamento dos construtos da LN seja feito de forma mais genérica.

A forma lógica numa ILN para acesso a banco de dados geralmente está relacionada com o esquema conceitual. Em geral as *queries* para um banco de dados podem ser mapeadas num subconjunto do esquema conceitual associado. Chen [1990] demonstrou que é possível mapear certos construtos da LN num esquema MER. Ele descreveu 11 regras de tradução para esquemas de bancos de dados em termos do diagrama MER associado. Numa extensão deste trabalho, Tseng, Chen e Yang [1992/93] propuseram um mapeamento das *queries* em LN numa álgebra relacional através da representação MER. A proposta de tradução para uma fórmula lógica intermediária no AGRON é fundamentada no processo de mapeamento de Tseng, Chen e Yang.

O processo de mapeamento usando o MER de Tseng, Chen eYang pressupõe uma análise prévia sintática-semântica, onde, assim como no sistema AGRON, os constituintes da sentença são associados a casos semânticos. Estes devem ser mapeados em estruturas da forma lógica para, posteriormente, serem traduzidos na linguagem de acesso ao banco de dados. Antes de iniciar o processo de mapeamento, é necessário que a análise da sentença em LN tenha fornecido:

- a estrutura de casos, onde o valor de cada caso é composto de um substantivo *cabeça* e seus modificadores
- o verbo relacionado à estrutura de casos

A essência do processo de mapeamento é a seguinte:

- cada caso é mapeado numa relação de entidades
- os substantivos cabeça e modificadores são mapeados nos atributos correspondentes de cada relação de entidades
- o verbo é mapeado numa relação de relacionamentos que associa as entidades envolvidas nos casos presentes

Um tratamento especial é dado aos verbos que não transferem ação, ou seja, os verbos de ligação e os verbos no imperativo.

Para cada relação de entidades, os atributos ou são identificados como o substantivo cabeça ou com os modificadores deste substantivo. Por exemplo, no MER do BMA, o atributo que corresponde ao substantivo cabeça na relação *prodpam* é *nome_produto*, e os atributos restantes correspondem a modificadores do produto. Na verdade, as relações de entidades representam os sintagmas nominais numa sentença em LN.

O dicionário tem um papel central, como também ocorre com o dicionário do AGRON. Em ambos os sistemas, é no dicionário que está armazenado o conhecimento linguístico que permite que os casos semânticos sejam mapeados nos atributos corretos das relações do banco de dados. Além disso, o dicionário possui informações sobre sinônimos, valores do domínio, e outros dados relevantes no processo de análise e tradução dos lexemas.

Assim sendo, o acesso ao dicionário permite que o verbo principal e os sintagmas nominais numa *query* sejam mapeados, respectivamente, para o relacionamento e as entidades correspondentes, e os valores dos casos semânticos, nos atributos associados. Logo, é a informação contida nas entradas léxicas associadas a cada terminal da sentença que orienta em grande parte o mapeamento.

A dificuldade maior no processo de tradução baseado no MER reside no fato de que, para um esquema geral, costumam existir diversos caminhos ou *paths* entre duas relações de entidades, o que pode dificultar ou impossibilitar a escolha do caminho correto. Decidir o caminho que corresponde ao relacionamento desejado muitas vezes é uma decisão que só pode ser tomada pelo próprio usuário.

A forma lógica usada para representar o resultado do mapeamento da sentença no MER associado é, na verdade uma extensão do próprio MER. Neste caso, a forma lógica considera os seguintes aspectos:

- a representação de modificadores;
- a representação do quantificador "todo"
- a representação das formas afirmativa e negativa dos verbos
- a representação das conjunções "e" e "ou"

Nesta forma lógica, a representação da sentença pode ser vista numa forma gráfica semelhante ao próprio MER. A notação é então acrescida de algumas figuras a fim de atender outros requisitos da forma lógica.

Os predicados associados aos atributos das relações são da forma "atributo θ constante", onde $\theta \in \{>, <, =, \neq, \geq, \leq\}$. Estes predicados são representados por nós ovais na forma lógica. Além disso, define-se o chamado *pseudo predicado* na forma de "atributo = ?" para representar o atributo alvo que deve ser exibido para o usuário.

Um retângulo sombreado é usado para representar a relação de entidades cujo valor do caso semântico é precedido pelo quantificador "todo".

Os verbos são associados aos losangos que representam relacionamentos. A forma negativa dos verbos é representada também como losangos, exceto que existe um triângulo na extremidade do losango que conecta a relação de entidades que *não* satisfaz o relacionamento.

A representação das conjunções é bastante simples, é feita através de linhas rotuladas com os símbolos $\wedge$ ou $\vee$ ligando os atributos, entidades ou relacionamentos envolvidos. Porém, o mapeamento das conjunções é um processo mais complexo. A conjunção "e" é, muitas vezes, interpretada como uma interseção de conjuntos, como em:

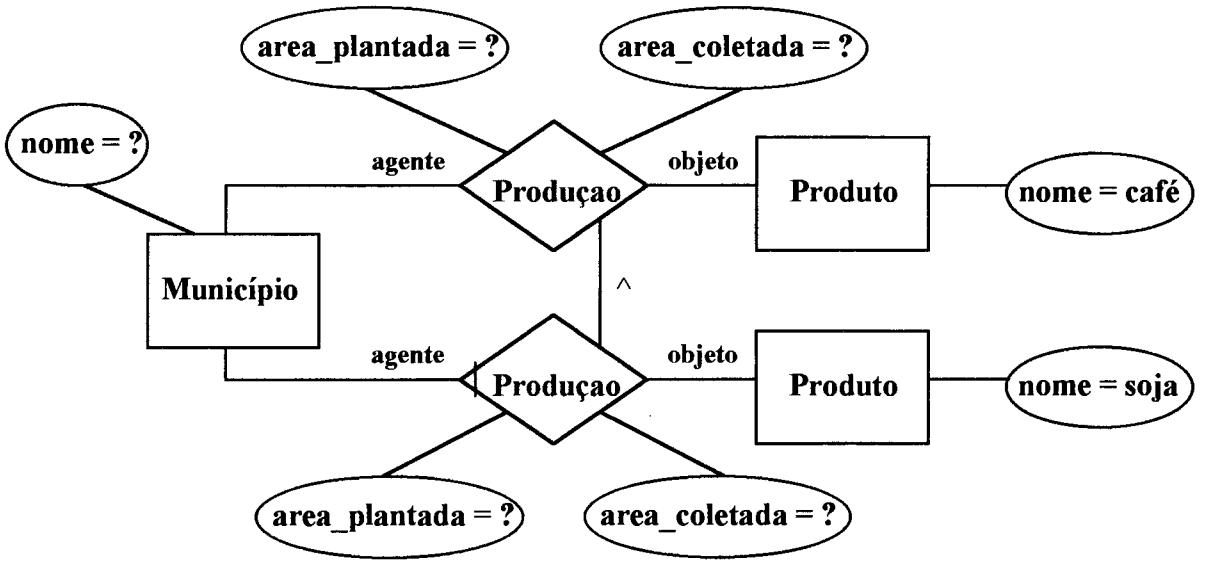
"Liste os produtos da lavoura temporária e os da lavoura permanente."

No sentido lógico, o "e" desta sentença se transforma num "ou", isto é, numa *disjunção lógica*. Tseng, Chen e Yang estudaram três situações onde as conjunções "e" e "ou" ocorrem ligando entidades e relacionamentos num MER, e analisaram o valor lógico das conjunções em cada situação. As três situações refletem, respectivamente, conjunções entre modificadores (modificador-modificador), entre relações de entidades (entidade-entidade) e entre relações de relacionamento (relacionamento-relacionamento). Somente no segundo tipo de conjunção, isto é, conjunções de sintagmas nominais que correspondem a entidades, pode ocorrer a transformação do "e" da sentença em LN para o "ou" lógico da forma lógica. Neste caso, a transformação ocorre se existem *pseudo predicados* ligados aos atributos das entidades, o que indica que o usuário deseja na verdade saber ambas as respostas.

Não discutiremos aqui a definição formal da forma lógica de Tseng, Chen e Yang, criada a partir desta extensão do MER, mas somente exemplificaremos a notação gráfica para tal forma lógica.

A seguir temos a forma lógica da sentença:

"Liste a área plantada e a área colhida nos municípios que produzem café e não produzem soja."



Forma lógica para a sentença: "Liste a área plantada e a área colhida nos municípios que produzem café e não produzem soja."

Outras formas intermediárias podem ser utilizadas para representar a sentença em LN. Por exemplo, no sistema TEAM [Grosz et al, 1985] a forma lógica é uma lógica de primeira ordem estendida, no CHAT [Warren & Pereira - 1980] é expressa em DCW, e no QPROC [Wallace, 1984] a forma lógica D&Q é uma estrutura hierárquica formada de descrições e qualificações.

A dificuldade, no entanto, é encontrar uma forma lógica que não exija muito esforço para tradução na forma final para acesso ao banco de dados. A forma lógica sugerida por Tseng, Chen e Yang, que estende a representação do MER, além de ser uma teoria bem recente, nos parece uma forma adequada no sentido de reduzir o trabalho de tradução para forma final numa linguagem de *query*, já que esta reflete os componentes do esquema conceitual associado ao banco de dados.

XIII. CONCLUSÃO

Propostas de Extensão

Assim como acontece em muitas ILNs, a primeira versão implementada da interface AGRON transforma a sentença em Linguagem Natural diretamente nos comandos da linguagem de *query* ao banco de dados, e, como na maioria dos casos, a justificativa principal para fazê-lo é obter uma performance melhor, além da facilidade para tratar certos aspectos da linguagem, tal como elipses. Porém, estes sistemas geralmente exigem um esforço considerável para serem adaptados a outros domínios semânticos, ou seja, não são propriamente transportáveis.

Neste sentido, a interface AGRON pode ser considerada uma interface *orientada ao domínio* (*domain-oriented*), pois não depende de um domínio específico, mas é projetada para o que denominamos *domínios espaço-temporais*, e pode ser facilmente adaptada a tais domínios. Como pudemos observar, os domínios espaço-temporais constituem informações organizadas sobre um determinado objeto segundo a disposição numa dimensão espaço-tempo.

O domínio espaço-temporal que utilizamos foi o banco de dados BMA da produção agrícola e pecuária no IBGE, onde estão armazenadas informações sobre produtos agropecuários nos diversos municípios brasileiros ao longo dos anos. Na verdade, a grande parte dos bancos de dados que integram o acervo do IBGE tem este mesmo caráter, pela própria natureza das informações produzidas pela instituição. Assim, outros exemplos de domínios espaço-temporais no IBGE são:

- o histórico dos índices do custo de vida nas regiões metropolitanas ao longo dos meses;
- a taxa de desmatamento nas regiões florestais no decorrer dos anos;
- o acompanhamento mensal dos preços de produtos diversos nas cidades brasileiras;
- o acompanhamento anual da poluição ambiental nas áreas industriais;
- a taxa de crescimento populacional nas principais cidades ao longo dos anos;
- o consumo mensal *per capita* de alimentos nos municípios.

Exemplos de outros universos semânticos não pesquisados no IBGE que se enquadram na estrutura espaço-temporal da interface AGRON ilustram o alcance das aplicações:

- a catalogação paleoantropológica de fósseis dos ancestrais da espécie humana em tempos pré-históricos, nos primórdios da civilização, nas diversas partes do planeta;

- o acompanhamento periódico dos investimentos e do consumo de energia nas principais cidades do mundo.

Estes são apenas alguns dos muitos exemplos de domínios facilmente adaptáveis ao escopo da interface aqui descrita, que se torna, na verdade, uma máscara semântica abrangente para um sem número de bancos de dados espaço-temporais. Neste caso, é necessário que exista sempre um objeto, ou tema principal a ser pesquisado, com uma determinada periodicidade, e ocorrendo num espaço qualquer, permitindo assim o mapeamento nos casos semânticos *objeto*, *espaço* e *tempo* definidos na gramática. Assim, a estrutura de casos da gramática, aliada a tradução discutida anteriormente para uma forma lógica intermediária segundo o MER, viabilizam a extensão da interface em dois sentidos: para outros domínios semânticos e para outras formas finais da sentença de acordo com o *back end* a ser desenvolvido.

A adaptação depende basicamente da atualização do dicionário, pois este contém todas as informações sintático-semânticas necessárias ao processamento da sentença. A atualização do dicionário pode ser feita por um especialista que conheça o sistema, através do acréscimo dos lexemas correspondentes as expressões que denotam objetos, modificadores, indicadores de local e época. Deixamos como recurso que visa a independência de especialistas, a proposta de desenvolvimento de uma ferramenta de apoio que, através de perguntas e respostas simples, possa interagir com um usuário inexperiente para obter as informações necessárias à manutenção do léxico. Na verdade, este recurso é utilizado em diversas ILNs, tal como ocorre no sistema TEAM.

Além disso, com a incorporação de novos lexemas, pode ser necessário acrescentar algumas construções linguísticas não previstas inicialmente na gramática. Por isso, o ideal é a implementação de um algoritmo de análise guiado por tabela (*table-driven*), que utilize alguma ferramenta para gerar automaticamente a tabela do analisador, o que diminui consideravelmente o impacto gerado pelas eventuais extensões da gramática. Da mesma forma que antes, estas novas regras poderiam ser identificadas por um sistema que estabelecesse um diálogo com o usuário, a fim de testar o valor semântico das construções.

Na verdade, o que pretendemos é propor um tipo de interface efetivamente transportável. Uma interface transportável deve possuir informação necessária para estabelecer uma ligação automática entre a maneira que o usuário pensa e a forma pela qual a informação é estruturada para processamento.

As dificuldades são evidentes, e a maior delas talvez seja o fato de os bancos de dados utilizarem formas de representação diferentes. Embora a entrada da interface seja de um só tipo, isto é, uma sentença em LN, a saída da interface deve

ser diferente dependendo da representação específica para cada banco de dados. A interface deve efetuar as transformações necessárias tornando transparentes ao usuário as particularidades do banco de dados em questão.

Para tanto, é necessário que haja um modelo que expresse as informações sobre os objetos do domínio do banco de dados, suas características e relacionamentos e as palavras e frases usadas para referenciá-los. O sistema deve então saber a relação entre as entidades deste modelo e a informação no banco de dados. Além disso, para garantir a transportabilidade é preciso distinguir, em todos os níveis da interface, as regras que são gerais e aquelas que são específicas do domínio, para que a gramática e o interpretador semântico não precisem ser reescritos; em outras palavras, é necessário a tradução para uma forma lógica intermediária que seja independente da aplicação. O uso de uma forma lógica que reflita o modelo conceitual parece ser uma representação adequada.

Verificamos assim que a transportabilidade da interface exige alguns pré-requisitos:

- o analisador sintático deve utilizar uma gramática geral ao invés de um conjunto de regras específicas do domínio do banco de dados;
- já que a interpretação semântica não pode ser pré-definida, é preciso desenvolver um mecanismo para adquirir semântica através de uma variedade de construções semânticas utilizando palavras e frases relacionadas ao domínio;
- as informações léxicas também devem ser adquiridas ou adaptadas de acordo com a gramática e com os conceitos do usuário do sistema.

Uma característica adicional desejável é que a adaptação da interface a novos domínios possa ser feita por alguém que não seja conhecedor da arquitetura interna do sistema e que não possua conhecimentos específicos das técnicas de processamento de Linguagem Natural.

A aquisição de semântica para os verbos é um realmente a maior dificuldade encontrada nas interfaces transportáveis. Na verdade, uma interface que não possua um dispositivo para adquirir a semântica dos verbos não pode ser considerada uma interface realmente transportável.

Por exemplo, na interface transportável TEAM, o processo de aquisição de verbos é uma sessão de perguntas e respostas ao DBE (especialista em banco de dados) visando:

- fornecer os tempos do verbo;
- fornecer uma sentença declarativa "mais geral possível" utilizando o verbo; a

partir desta sentença o TEAM extrai informações sobre a transitividade do verbo, definindo um predicado para o mesmo e classificando seus argumentos;
- avaliar possíveis construções utilizando o verbo, através da exibição de sentenças corretas ou não.

Além destes recursos, sugerimos uma sofisticação nas formas de representação do conhecimento utilizadas. Na presente implementação, toda a informação semântica está codificada no dicionário e nas regras da gramática. Porém, para aumentar ainda mais o poder expressivo da linguagem, seria necessário acoplar uma rede semântica, tal como a taxonomina de subsunções discutida inicialmente, para permitir a análise semântica de outras construções linguísticas que hoje não podem ser processadas. Por exemplo, considere a sentença:

"Quais os tipos de produto existentes?"

Esta pergunta só pode ser respondida conhecendo-se a natureza dos produtos no banco de dados, que é uma informação que não pode ser armazenada diretamente no dicionário conforme está definido atualmente, mas necessita de um formalismo mais poderoso para ser representada. Neste caso, o banco de dados pode armazenar diversos tipos de produto, por exemplo, produtos agrícolas, da pecuária, da indústria, do comércio, e outros, que estabelecem classificações e relações mais complexas que só podem ser descritas em representações mais sofisticadas como redes semânticas.

Resultados

Embora o sistema AGRON tenha sido projetado para atender necessidades reais de disseminação de informações no IBGE, não foi possível testar a utilização do mesmo junto aos usuários finais, devido a prioridades que foram estabelecidas em função do trabalho de atendimento às frequentes solicitações externas, além de problemas com a manutenção de sistemas ora existentes.

Realizamos testes com um usuário inexperiente que formulou cerca de 80 (oitenta) enunciados a fim de podermos avaliar o grau de adequação da interface a situações reais. O usuário recebeu uma descrição sobre as informações contidas no banco de dados da produção agrícola, e sobre as formas e construções linguísticas que poderiam ser usadas no AGRON. Além disso, executamos uma demonstração através de uma sessão com uma sequência de 10 (dez) sentenças em LN para que o indivíduo se familiarizasse com a dinâmica do sistema.

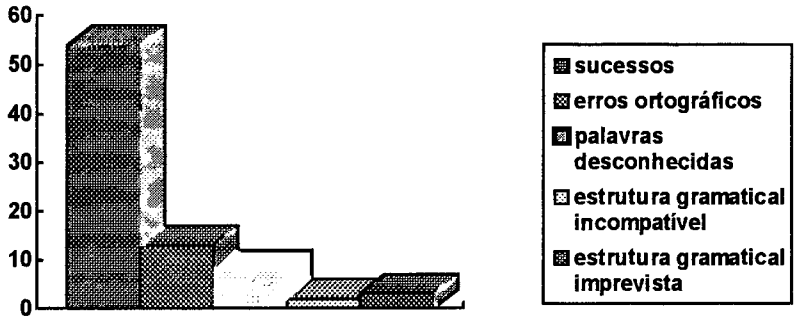
Os testes evidenciaram que cerca de 68% das sentenças formuladas pelo usuário foram inteiramente processadas pelo sistema, fornecendo uma tradução SQL associada. Em 14% destes casos, porém, a tradução não correspondia as intenções do usuário. Estas incorreções na tradução geralmente estavam ligadas a incapacidade de identificar o escopo de certas conjunções e disjunções, e ao fato de não processarmos certas cláusulas do SQL tais como o *subselect*, por exemplo.

No restante dos enunciados, ou seja, em 32% das vezes, as sentenças não puderam ser processadas, e apresentaram erros provenientes de quatro situações diferentes:

1) erros ortográficos;
2) palavras que não foram incorporadas a linguagem;
3) estrutura sintagmal não compatível com a gramática adotada;
4) estrutura sintagmal compatível com a gramática, porém imprevista.

Além disso, pelo fato da complexidade do algoritmo de análise ser linear - *O(n)* no comprimento das *n* palavras que compõem a entrada - o tempo de resposta foi considerado ótimo. A seguir fornecemos um histograma com os resultados condensados.

Variações obtidas no processamento de 80 sentenças em LN no AGRON



Um teste mais complexo deveria efetuar uma avaliação mais abrangente da interface em termos de aplicabilidade e eficiência em relação a outros sistemas. Propomos um experimento semelhante ao descrito no recente estudo de Bell & Rowe [1992], que consiste numa análise exploratória do comportamento de três estilos diferentes de interface para *query* a banco de dados: interfaces artificiais (baseadas numa linguagem formal), interfaces gráficas e interfaces em linguagem natural. Todas as três interfaces utilizadas eram produtos comerciais. O estudo revelou pontos fortes e fracos de cada interface e mostrou diferenças quantitativas e qualitativas entre cada uma delas.

O estudo de Bell & Rowe é considerado exploratório devido a ausência, até então, de pesquisas deste tipo, que comparassem estes três estilos de interface através de sistemas reais. A pesquisa difere de outras anteriores em três aspectos:

1) é a primeira a incluir uma interface gráfica;
2) foi realizada num ambiente relativamente controlado usando produtos reais;
3) enfatiza os dados qualitativos a fim de auxiliar na compreensão da natureza da comparação entre os três sistemas.

Daremos um panorama de como o estudo de Bell & Rowe foi realizado. O banco de dados utilizado incluía informações sobre estudantes, professores, classes e atividades para uma escola de segundo grau. A escolha foi feita em função do fato de ser este um banco de dados familiar a todos os indivíduos, simples de compreender, além de apresentar um número de entidades e relacionamentos suficientes para permitir uma variedade de perguntar simples e complexas.

Desenvolveu-se uma representação pictórica das tarefas a fim de minimizar o problemas de como apresentar as tarefas aos indivíduos de maneira justa, sem conduzí-los a um ou outro tipo de resposta.

As três interfaces consideradas foram:

- a linguagem artificial SQL;
- a interface gráfica Simplify;
- a interface em linguagem natural DataTalker.

O experimento foi realizado com 55 (cinquenta e cinco) indivíduos com níveis diferentes de experiência computacional, que foram divididos em quatro grupos: novatos, usuários finais, programadores e especialistas em banco de dados. Cada grupo consistia de quinze indivíduos, exceto os especialistas em banco de dados, uma vez que não fazia sentido submetê-los ao teste com o uso de SQL, pois eles possuíam conhecimento desta linguagem. Assim, somente dez usuários de banco de dados foram incluídos no estudo.

As tarefas variavam de questões simples (usando uma única tabela do banco de dados) até questões complexas (envolvendo múltiplas tabelas e agregações). As tarefas foram divididas em duas fases: a fase de aprendizado e a fase de desempenho. A fase de aprendizado era composta de níveis de tarefas que cobriam sete tipos de *queries*, conforme o quadro a seguir.

N1	uma tabela, nenhuma restrição
N2	uma tabela, uma restrição
N3	uma tabela, duas restrições
N4	uma tabela, uma ou duas restrições, ordenação
N5	uma tabela, nenhuma restrição, agregação na contagem
N6	junção de duas tabelas
N7	junção de três tabelas

Definição de níveis de tarefa na fase de aprendizado

Nesta fase, os indivíduos recebiam ajuda, que era fornecida em quatro etapas, até poderem executar as tarefas com sucesso. Quando o indivíduo podia resolver as tarefas sem ajuda, então passava ao nível seguinte.

A fase de desempenho incluía questões para testar o quanto o usuário aprendeu a usar a interface a partir da fase de aprendizado. Nesta fase os usuários não recebiam nenhum tipo de ajuda. Além disso, foram acrescentadas questões para testar se o indivíduo era capaz de estender seu conhecimento da interface a situções que não tinham sido explicitamente ensinadas. O quadro a seguir mostra os tipos de tarefa nesta fase.

1	uma tabela, nenhuma restrição	N1
2	uma tabela, duas restrições	N2 e N3
3	uma tabela, agregação	N5
4	junção de três tabelas	N7
5	junção de três tabelas, uma restrição, ordenação	N4 e N6
6	não-existência	nenhum
7	junção auto-referencial	nenhum

Níveis de tarefa na fase de desempenho

Cada indivíduo recebeu uma breve introdução ao experimento, ao banco de dados e a representação pictórica de tarefas, trabalhando cerca de duas horas com a interface designada.

Os resultados quantitativos foram baseados no exame de duas medidas: sucesso e média de tempo para resolução da tarefa. O sucesso na execução da tarefa visava medir o quanto os indivíduos eram capazes de executar uma variedade de tarefas depois do treinamento. O tempo refletia o quanto de trabalho era necessário para completar tais tarefas.

Algumas constatações e análises sobre o **sucesso** na fase de desempenho foram:

- O segundo nível causou problemas para todos os indivíduos que utilizaram o DataTalker. Esta tarefa incluía múltiplas restrições cuja especificação pode ser diversificada e um tanto complexa na linguagem natural.

- Os novatos utilizando DataTalker não obtiveram mais sucesso do que aqueles usando SQL ou Simplify, apesar dos benefícios esperados das interfaces em linguagem natural.

- No quinto nível, os usuários finais utilizando DataTalker obtiveram um sucesso muito maior do que aqueles usando SQL ou Simplify. Este sucesso se deve, provavelmente, ao fato dos usuários do DataTalker não necessitarem lidar com certas complexidades decorrentes da interação com os outros sistemas neste tipo de tarefa.

- As taxas de sucesso foram semelhantes nos outros níveis 1, 3 e 4 de tarefas treinadas para os três tipos de interface.

- As questões sobre não-existência foram executadas com sucesso pela maioria dos usuários do DataTalker, ao passo que os indivíduos usando SQL ou Simplify não obtiveram êxito absolutamente. Este resultado foi de fato um ponto positivo para a interface em linguagem natural, especialmente no que

concerne ao processamento de tarefas não treinadas.

- O outro tipo de tarefa não treinada, a junção auto-referencial, só foi efetuada por um único indivíduo, um usuário de banco de dados usando Simplify. Este resultado sugere que os indivíduos realmente não conseguiram compreender o conceito de auto-referência ou auto-relacionamento e/ou a forma que este deve ser especificado.

A média de **tempo** na execução das tarefas não revelou grandes diferenças, o que não implica que não existam dificuldades relevantes no uso de uma ou outra interface. É importante lembrar que as avaliações apresentadas são interpretações de resultados quantitativos baseados nas observações qualitativas feitas pelos autores, e não devem ser consideradas como conclusões baseadas em números significativos estatisticamente. Os próprios autores salientam que é necessário mais experimentação e análise.

Os resultados qualitativos neste estudo são interessantes porque estabelecem certos parâmetros de comparação entre as interfaces. Neste caso, os resultados obtidos abordam, entre outros aspectos, os seguintes tópicos:

- Restrição única e restrições múltiplas - A transição de uma única restrição para múltiplas restrições resultou em desempenhos notadamente distintos em cada uma das três interfaces. Em particular, esta transição se mostrou mais árdua para os usuários do DataTalker devido, em parte, às muitas maneiras possíveis de se especificar múltiplas restrições usando a linguagem natural.

- Junções - Ambos os usuários de SQL e Simplify precisaram compreender o conceito de junção de tabelas, o que se mostrou um processo complexo devido a identificação dos termos para junção (que deveriam ser igualados) em cada tipo de tarefa. Por outro lado, o DataTalker foi claramente superior, pois os usuários não precisavam absolutamente especificar as junções, nem mesmo percebiam que estas ocorriam. De fato, alguns indivíduos pensaram que as tarefas relativas a junção de tabelas já haviam sido dadas anteriormente, pois, para eles, estas tarefas não diferiam de outras anteriores.

- Obter a resposta errada sem saber - Esperava-se que os usuários do DataTalker apresentassem um índice maior de respostas erradas desapercebidas. Contudo, este comportamento ocorreu com praticamente a mesma frequência em todos os sistemas.

As conclusões gerais de Bell & Rowe sobre tudo o que foi observado durante o experimento se resumem em três pontos principais:

1. Não existe a melhor interface. Nenhuma das três interfaces se destacou sobre as outras em todos os casos, mas apenas sob certas condições específicas.

2. A interação com o DataTalker foi diferente daquela com o SQL e Simplify. Este resultado aponta um objetivo implícito das Interfaces em Linguagem Natural, ou seja, mudar o estilo de interação homem-máquina. Além disso, a performance geral do DataTalker foi comparável a das interfaces estáveis.

3. A experiência do usuário afeta diferentemente o desempenho em cada um dos tipos de interface. A performance no DataTalker foi afetada pela experiência em computação e em banco de dados. Geralmente, assume-se que as ILNs sejam melhores para aqueles com pouca experiência computacional, mas parece que tal conhecimento ainda é importante para os usuários das interfaces de hoje. Por outro lado, o desempenho com o Simplify foi fortemente afetado pela experiência em programação. Neste caso também esperava-se que as interfaces gráficas fossem mais para úteis àqueles com menos experiência.

Embora estes resultados apliquem-se especificamente às interfaces testadas, espera-se que a extensão dos mesmos à outras interfaces seja aplicável devido a semelhança com outros sistemas existentes. Contudo, se o mesmo teste for realizado com indivíduos brasileiros, há que se levar em conta o nível de desconhecimento do inglês, que provavelmente afetará significativamente para pior o desempenho em qualquer das interfaces utilizadas. Por outro lado, o uso de interfaces dos três tipos voltadas para o português possibilitaria uma análise mais adequada e uma validação, ou não, dos resultados obtidos por Bell & Rowe.

Para possibilitar comparações mais significativas, sugere-se ainda uma pesquisa mais ampla enfatizando outros aspectos qualitativos tais como a avaliação das estratégias utilizadas pelo usuário quando trabalhando numa ILN, ou um modelo de conhecimento e comportamento dos usuários no uso de interfaces de *query* em geral. Além disso, é importante considerar outros tipos de interface tais como os sistemas em linguagem natural baseados em menus (*NLMenu systems*).

Contribuições

Neste trabalho descrevemos o desenvolvimento e implementação de um protótipo da interface AGRON, voltada para comandos e perguntas dirigidas a bancos de dados históricos com estatísticas sobre a produção agrícola. Uma das características do sistema é que ele pode ser facilmente adaptado a outros domínios espaço-temporais de acordo com a estrutura adotada de temas semânticos.

Neste contexto, destacamos as técnicas de utilização de gramáticas semânticas baseadas numa estrutura de casos que contempla as dimensões espaço-tempo e um esquema para processamento de cláusulas relativas com restrições de busca.

Outras facilidades incluem o tratamento de ambiguidades, de movimentos não-locais ou arbitrários entre os casos semânticos, a hierarquia de tipos do caso espacial, a criação de classes semânticas e o uso de um léxico aumentado com informações semânticas, permitindo que o algoritmo de análise trabalhasse informações sintáticas e semânticas simultaneamente através da orientação para o sentido das palavras e expressões.

Além disso, enfatizamos o processamento de sentenças elípticas segundo a taxonomia de Frederking [1986], o que viabilizou a expansão da análise a um nível contextual, e simplificou a interação usuário-sistema, através da redução do tamanho dos enunciados.

Propomos extensões para um sistema transportável cuja forma final mapeia uma forma lógica que reflete o modelo conceitual do banco de dados. O desenvolvimento de ferramentas interativas para aquisição de semântica e manutenção do léxico completam o quadro da portabilidade e são vistas como meio de tornar a interface independente da adaptação realizada por um especialista conhecedor do sistema.

Também sugerimos o acoplamento de uma rede semântica de subsunções como forma de representação da estrutura conceitual dos objetos de pesquisa, a fim de aumentar a flexibilidade e a expressibilidade das construções linguísticas aceitas.

Apresentamos alguns resultados expressivos e avaliamos a possibilidade de aplicação de um estudo comparativo entre a ILN e outros dois tipos de interface - gráfica e artificial - com o objetivo de verificar a aplicabilidade da Linguagem Natural em interfaces para *query* a bancos de dados.

APÊNDICE A

Regras da Gramática

S → comd [objt] [espc] [temp] [rest] |
itrr [objt] [espc] [temp] [rest] |
(elip) [objt] [espc] [temp] [rest]

[espc] → ppre [detm] [emun] |
ppre [detm] [eunf] |
ppre [detm] [ereg] |
spre [emun] |
spre [eunf] |
spre [ereg] |
spre bras

[emun] → smun spre sunf spre unfd |
smun spre sunf unfd |
smun spre unfd |
smun spre sreg regi |
smun spre regi |
smun spre [lmun] |
smun [lmun] |
[lmun]

[eunf] → sunf spre sreg regi |
sunf spre regi |
sunf spre bras |
sunf spre [lunf] |
sunf [lunf] |
[lunf]

[ereg] → sreg spre bras |
sreg [lreg] |
[lreg]

[lmun] → munc [lmun] |
munc

[lunf] → unfd [lunf] |
unfd

[lreg] → regi [lreg] |
regi

[temp] → ppre artd sano [lano] |
ppre [lano] |
cpre artd sano [lano] |
cpre spre sano [lano] |
cpre [lano] |
spre sano [lano] |
spre [lano]

[lano] → numr dpre numr |
numr [lano] |
numr

[objt] → [detm] smun qual objs |
[detm] sunf qual objs |
[detm] sreg qual objs |
[detm] [emun] |
[detm] [eunf] |
[detm] [ereg] |
artd sano |
[artd] sobj |
[artd] sobj spre objc |
[lmod] objs |
[lmod] sobj spre objc

[lmod] → [artd] modf [lmod] |
[artd] modf

[detm] → artd |
quat |
card

[rest] → cond modf [lres]
cond verb [lres] objs
cond verb objs
verb [lres] objs
verb objs

[lres] → qpre numr cjun [lres] |
qpre numr djun [lres] |
qpre numr unmd |
mpre numr cjun [lres] |
mpre numr djun [lres]
mpre numr unmd

APÊNDICE B

Dicionário Fixo - Tabela de Gramemas

GRAMEMA	CLASSE	G	N
a	artigo definido	f	s
a	preposição	-	-
abaixo de	locução prepositiva	-	-
acima de	locução prepositiva	-	-
além de	locução prepositiva	-	-
algum	pronome indefinido	m	s
alguma	pronome indefinido	f	s
algumas	pronome indefinido	f	p
alguns	pronome indefinido	m	p
antes de	locução prepositiva	-	-
após	preposição	-	-
as	artigo definido	f	p
até	preposição	-	-
cada	pronome indefinido	i	i
cinco	numeral cardinal	-	-
com	preposição	-	-
cuja	pronome relativo	f	s
cujas	pronome relativo	f	p
cujo	pronome relativo	m	s
cujos	pronome relativo	m	p
da	artigo+ preposição	f	s
das	artigo+ preposição	f	p
de	preposição	-	-
depois de	locução prepositiva	-	-
desde	preposição	-	-
dez	numeral cardinal	-	-
diferente de	locução prepositiva	-	-
do	artigo+ preposição	m	s
dois	numeral cardinal	m	-
dos	artigo+ preposição	m	p
duas	numeral cardinal	f	-
e	conj. coord. aditiva	-	-
em	preposição	-	-
entre	preposição	-	-
igual a	locução prepositiva	-	-
inferior a	locução prepositiva	-	-
maior (do) que	locução prepositiva	-	-
mais (do) que	locução prepositiva	-	-
menor (do) que	locução prepositiva	-	-

menos (do) que	locução prepositiva	-	-
na	artigo+ preposição	f	s
nas	artigo+ preposição	f	p
no	artigo+ preposição	m	s
nos	artigo+ preposição	m	p
nove	numeral cardinal	-	-
o	artigo definido	m	s
oito	numeral cardinal	-	-
onde	advérbio interrog.	-	-
onde	pronome relativo	i	i
os	artigo definido	m	p
ou	conj. coord. alternat.	-	-
outra	pronome indefinido	f	s
outras	pronome indefinido	f	p
outro	pronome indefinido	m	s
outros	pronome indefinido	m	p
para	preposição	-	-
quais	pronome interrog.	i	p
qual	pronome interrog.	i	s
quando	advérbio interrog.	-	-
quanta	pronome interrog.	f	s
quantas	pronome interrog.	f	p
quanto	pronome interrog.	m	s
quantos	pronome interrog.	m	p
quatro	numeral cardinal	-	-
que	pronome interrog.	i	i
que	pronome relativo	i	i
seis	numeral cardinal	-	-
sete	numeral cardinal	-	-
superior a	locução prepositiva	-	-
toda	pronome indefinido	f	s
todas	pronome indefinido	f	p
todo	pronome indefinido	m	s
todos	pronome indefinido	m	p
três	numeral cardinal	-	-
um	numeral cardinal	m	-
uma	numeral cardinal	f	-

APÊNDICE C

Dicionário Móvel - Tabela de Lexemas

LEXEMA	CLASSE	G	N
abacate	substantivo	m	s
abacaxi	substantivo	m	s
açúcar	substantivo	m	s
agrícola	adjetivo	f	s
alfafa fenada	sintagma nominal	f	s
algodão arbóreo	sintagma nominal	m	s
algodão herbáceo	sintagma nominal	m	s
alho	substantivo	m	s
amendoim	substantivo	m	s
ano	substantivo	m	s
anos	substantivo	m	p
apresentar	verbo	-	s
apresente	verbo	3	s
arbóreo	adjetivo	m	s
área coletada	sintagma nominal	f	s
área plantada	sintagma nominal	f	s
arroz	substantivo	m	s
arroz irrigado	sintagma nominal	m	s
arroz sequeiro	sintagma nominal	m	s
aveia	substantivo	f	s
azeitona	substantivo	f	s
baía	substantivo	f	s
banana	substantivo	f	s
batata doce	sintagma nominal	f	s
batata inglesa	sintagma nominal	f	s
borracha látex coagulado	sintagma nominal	f	s
borracha látex líquido	sintagma nominal	f	s
Brasil	substantivo	m	s
cacau	substantivo	m	s
cacho	substantivo	m	s
cachos	substantivo	m	p
café	substantivo	m	s
cajú	substantivo	m	s
cana forrageira	sintagma nominal	f	s
cana-de-açúcar	sintagma nominal	f	s
caquí	substantivo	m	s
castanha-de-cajú	sintagma nominal	f	s
cebola	substantivo	f	s
centeio	substantivo	m	s

cevada	substantivo	f	s
chá-da-índia	sintagma nominal	m	s
cidade	substantivo	f	s
cidades	substantivo	f	p
coagulado	adjetivo	m	s
côco-da-baía	sintagma nominal	m	s
coletada	adjetivo	f	s
colhe	verbo	3	s
colhem	verbo	3	p
colheram	verbo	3	p
colheu	verbo	3	s
colhida	adjetivo	f	s
Cr$	abreviatura de substantivo	m	p
cruzeiros	substantivo	m	p
cultiva	verbo	3	s
cultivada	adjetivo	f	s
cultivam	verbo	3	p
cultivaram	verbo	3	p
cultivo permanente	sintagma nominal	m	s
cultivo temporário	sintagma nominal	m	s
cultivos permanentes	sintagma nominal	m	p
cultivos temporários	sintagma nominal	m	p
cultivou	verbo	3	s
cultura permanente	sintagma nominal	f	s
cultura temporária	sintagma nominal	f	s
culturas permanentes	sintagma nominal	f	p
culturas temporárias	sintagma nominal	f	p
dendê	substantivo	m	s
doce	adjetivo	m	s
é	verbo	3	s
era	verbo	3	s
eram	verbo	3	p
erva-mate	sintagma nominal	f	s
ervilha	substantivo	f	s
estado	substantivo	m	s
estados	substantivo	m	p
exiba	verbo	3	s
exibir	verbo	-	s
fava	substantivo	f	s
federação	substantivo	f	s
feijão	substantivo	m	s
fenada	adjetivo	f	s
figo	substantivo	m	s
foi	verbo	3	s
foram	verbo	3	p

forneça	verbo	3	s
fornecer	verbo	-	s
forrageira	adjetivo	f	s
fruto	substantivo	m	s
frutos	substantivo	m	p
fumo	substantivo	m	s
goiaba	substantivo	f	s
granífero	adjetivo	m	s
guaraná	substantivo	m	s
hectare	substantivo	m	s
hectares	substantivo	m	p
herbáceo	adjetivo	m	s
imprima	verbo	3	s
imprimir	verbo	-	s
índia	substantivo	f	s
informar	verbo	-	s
informe	verbo	3	s
inglesa	adjetivo	f	s
irrigado	adjetivo	m	s
juta	substantivo	f	s
laranja	substantivo	f	s
látex	substantivo	m	s
lavoura permanente	sintagma nominal	f	s
lavoura temporária	sintagma nominal	f	s
lavouras permanentes	sintagma nominal	f	p
lavouras temporárias	sintagma nominal	f	p
limão	substantivo	m	s
linho	substantivo	m	s
líquido	adjetivo	m	s
listar	verbo	-	s
liste	verbo	3	s
maçã	substantivo	f	s
malva	substantivo	f	s
mamão	substantivo	m	s
mamona	substantivo	f	s
mandioca	substantivo	f	s
manga	substantivo	f	s
maracujá	substantivo	m	s
marmelo	substantivo	m	s
mate	substantivo	m	s
médio	adjetivo	m	s
melancia	substantivo	f	s
melão	substantivo	m	s
milho	substantivo	m	s
mostrar	verbo	-	s

mostre	verbo	3	s
município	substantivo	m	s
municípios	substantivo	m	p
noz	substantivo	f	s
obtenha	verbo	3	s
obter	verbo	-	s
palmito	substantivo	m	s
pêra	substantivo	f	s
permanente	adjetivo	i	s
permanentes	adjetivo	i	p
pêssego	substantivo	m	s
pimenta-do-reino	sintagma nominal	f	s
plantada	adjetivo	f	s
plantam	verbo	3	p
plantaram	verbo	3	p
plantou	verbo	3	s
preço médio	sintagma nominal	m	s
produção	substantivo	f	s
produto	substantivo	m	s
produtor	adjetivo	m	s
produtora	adjetivo	f	s
produtoras	adjetivo	f	p
produtores	adjetivo	m	p
produtos	substantivo	m	p
produz	verbo	3	s
produzem	verbo	3	p
produzida	adjetivo	f	s
produziram	verbo	3	p
produziu	verbo	3	s
quantidade produzida	sintagma nominal	f	s
rami	substantivo	m	s
região	substantivo	f	s
regiões	substantivo	f	p
reino	substantivo	m	s
são	verbo	3	p
seja	verbo	3	s
sejam	verbo	3	p
semeada	adjetivo	f	s
semearam	verbo	3	p
semeia	verbo	3	s
semeiam	verbo	3	p
semeou	verbo	3	s
sequeiro	adjetivo	m	s
sisal	substantivo	m	s
soja	substantivo	f	s

sorgo	substantivo	m	s
tangerina	substantivo	f	s
tem	verbo	3	s
temporária	adjetivo	f	s
temporárias	adjetivo	f	p
temporário	adjetivo	m	s
temporários	adjetivo	m	p
tenha	verbo	3	s
tenham	verbo	3	p
teve	verbo	3	s
tiveram	verbo	3	p
tomate	substantivo	m	s
tonelada	substantivo	f	s
toneladas	substantivo	f	p
total	substantivo	m	s
trigo	substantivo	m	s
tungue	substantivo	m	s
uf	abreviatura de sint. nominal	m	s
ufs	abreviatura de sint. nominal	m	p
unidade	substantivo	f	s
unidades	substantivo	f	p
urucum	substantivo	m	s
uva	substantivo	f	s
valor	substantivo	m	s

BIBLIOGRAFIA

- **Aho**, A.; **Sethi**, R.; **Ullman**, J.; "Compilers, Principles, Techniques, and Tools"; 796 p.; Ed. Addison Wesley Publishing Company; 1986.
- **Allen**, J.; "Natural Language Processing"; 574 p.; Ed. The Benjamin/Cummings Publishing Company; 1987.
- **Alves**, E.; "Uma Aplicação em Redes Semânticas utilizando Redes Bayesianas"; COPPE/UFRJ, M. Sc., Engenharia de Sistemas e Computação; 1993.
- **Appelt**, D. E.; **Martin**, P. A.; **Pereira**, F. N.; "Transportability and Generality in a Natural Language Interface System"; IJCAI, p. 573-581; 1983.
- **Bates**, M.; **Moser**, M.; **Stallard**, D.; "The IRUS Transportable Natural Language Database Interface"; Expert Database Systems, p.617-638; Ed. Larry Kerschberg; 1986.
- **Bell**, J. E.; **Rowe**, L. A.; "An Exploratory Study of Ad Hoc Query Languages to Databases"; Proceedings of IEEE, p. 606-613; 1992.
- **Bernorio**, M.; **Bertoni**, M.; **Dabbene**, A.; **Solmavico**, M.; "Querying Databases with a Domain Oriented Natural Language Understanding System"; Journal of Computer and Information Sciences, Vol. 9, No. 2, p. 141-159; 1980.
- **Coelho**, H.; "Man-Machine Communication in Portuguese: A Friendly Library Service System"; Inform. Systems, Vol. 7, No. 2, p. 163-181; Pergamon Press Ltd; 1982.
- **Euzenat**, B.; **Normier**, B.; **Ogonowski**, A.; "SAPHIR+RESEDA, A New Aproach to Intelligent Data Base Access"; IJCAI, p. 855-857; 1985.
- **Finin**, T.; **Kass**, R.; "Modeling the User in Natural Language Systems"; Computational Linguistics, Vol. 14, No. 3, p. 5-22; 1988.
- **Frederking**, R. E.; "Natural Language Dialogue in an Integrated Computational Model"; D. Sc., Department of Computer Science, Carnegie-Mellon University; 1986.
- **Groz**, B. J.; **Appelt**, D. E.; **Martin**, P. A.; **Pereira**, F. N.; "TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces"; Artificial Inteligence, No. 32, p. 173-244; 1987.
- **Guida**, G.; **Solmavico**, M.; **Comino**, R.; **Gemello**, R.; **Rullent**, C.; **Sisto**, L.; "Understanding Natural Language through Parallel Processing of Syntactic and Semantic Knowledge: An Application to Data Base Query"; IJCAI, p. 663-668; 1983.
- **Hendrix**, G. G.; **Sacerdoti,** E. D.; **Sagalowicz**, D. ; **Slocum**, J.; "Developing a Natural Language Interface to Complex Data"; ACM Transactions on Database Systems, Vol. 3, No. 2, p. 105-147; 1978.
- **IBM Corporation**; "IBM Database 2, Version 2; SQL Reference, Release 2"; 1989.

- **IBM Corporation**; "IBM Database 2, Version 2; SQL User's Guide, Release 1"; 1988.
- **Jones**, K. S.; **Boguraev**, B. K.; "How to Drive a Database Front End Using General Semantic Information"; Proc. Conf. on Applied Natural Language Processing, Santa Monica, p. 81-88; 1983.
- **Kaplan**, S. J.; "Co-operative Responses from a Portable Natural Language System"; Computational Models of Discourse, p. 167-208; Ed. M. Brady and B. Berwick; 1983.
- **Ott**, N.; "Aspects of the Automatic Generation of SQL Statements in a Natural Language Query Interface"; Information Systems Vol. 17, No. 2, p. 147-159; 1992.
- **Perrault**, C. R.; **Grosz**, B. J.; "Natural Language Interfaces"; Annual Review of Computer Science No.1, p. 47-82; 1986.
- **Ritchie**, G.; **Thompson**, H.; "Natural Language Processing", Artificial Intelligence, cap. 11, p. 358-388; 1983.
- **Sacerdoti**, E. D.; "Language Acess to Distributed Data with Error Recovery (LADDER)"; IJCAI, p. 196-202; 1977.
- **Templeton**, M.; **Burger**, J.; "Problems in Natural Language Interface to DBMS with Examples from EUFID"; Proc. Conf. on Applied Natural Language Processing, Santa Monica, p. 3-16; 1983.
- **Tseng**, F.; **Chen**, A.; **Yang**, W.; "On Mapping Natural Language Constructs into Relational Algebra thru E-R Representation"; Data & knowledge Engineering No. 9, p. 97-118; 1992/93.
- **Wallace**, M.; "Communicating with Databases in Natural Language"; 170 p.; Ed. Ellis Horwood Limited, 1984.
- **Weischedel**, R. M.; "Knowledge Representation and Natural Language Processing", Proceedings of the IEEE; 1986.
- **Winograd**, T.; "Language as a Cognitive Process"; Vol. I: Syntax, 640 p.; Ed. Addison -Wesley Publishing Company; 1983.
- **Woods**, W. A.; "Semantics and Quantification in Natural Language Question Answering"; Advances in Computers, Vol. 17, p. 2-87; Ed. M. Yovitz, Academic Press; 1978.
- **Woods**, W. A.; "Understanding Subsumption and Taxonomy: A Framework for Progress"; Principles of Semantic Networks, p. 45-94; Ed. Morgen Kaufman; 1991.