API PARA VALIDAÇÃO DE EDITORES MOLECULARES

Ronaldo Salvador*
Orientadora: Profa. Dra. Andréa Lucia Braga Vieira Rodrigues**

RESUMO: O projeto consiste em desenvolver uma API (Application Programming Interface) para validação de cadeias moleculares geradas por qualquer editor molecular que utilize conceitos como o SMILES (Simplified Molecular Input Line Entry Specification), representação linear para fórmulas estruturais.
A API utiliza análise léxica e sintática na sentença obtida através de um editor molecular, em busca de possíveis erros gramaticais na sentença. O algoritmo de validação usa o conceito de ligação covalente dos átomos para encontrar erros nas estruturas, sendo necessário para isso transformar a expressão regular SMILES em uma expressão numérica simples e a partir de regras predefinidas, criar uma seqüência de cálculos em busca de possíveis erros na estrutura molecular.

PALAVRAS-CHAVE: Editor molecular. Validação. SMILES. Algoritmo. Análise léxica. Análise sintática.

API FOR VALIDATION OF MOLECULAR EDITORS

ABSTRACT: The project is to develop an API (Application Programming Interface) for validation of molecular chains generated by any editor molecular that uses concepts such as SMILES (Simplified Molecular Input Line Entry Specification), linear representation for structural formulas.
The API uses syntactic and lexical analysis in the string obtained by a molecular editor, in search of possible grammatical errors. The algorithm uses the concept of validation covalent bonding of atoms to find errors in the structures and is required for that transform the regular expression SMILES in a simple numerical expression and from predefined rules, creates a sequence of calculations in search of possible errors in the structure Molecular.

KEY WORDS: Editor molecular. Validation. SMILES. Algorithm. Lexical analysis. Syntactic analysis.

* Graduado em Ciência da Computação pela Universidade de Sorocaba (2008). Tem experiência na área de Ciência da Computação.
Endereço: Rua Vicente Paes Filho, 505 CEP 18070-380 Sorocaba / SP. E-mail: ronaldosalva@yahoo.com.br
Apoio: FAPESP
** Graduada em Engenharia Elétrica pela Faculdade de Engenharia de Sorocaba (1990), mestrado em Engenharia Elétrica pela Universidade de São Paulo (1994) e doutorado em Engenharia Elétrica pela Universidade de São Paulo (2002). Atualmente é professor titular da Universidade de Sorocaba e Professor Titular / Coordenador da Faculdade de Engenharia de Sorocaba. Tem experiência na área de Engenharia Elétrica, com ênfase em Engenharia de Computação. Atuando principalmente nos seguintes temas: EDI - Intercâmbio Eletrônico de Dados, Comunicação via Internet, UN/EDIFACT, XML, Comunicação Estruturada e SED - Supervia Eletrônica de Dados.

Recebido em: Março / 2008 Avaliado em: Maio / 2008

Introdução

Editor molecular é um software que permite desenhar fórmulas estruturais de moléculas e é capaz de gerar sentenças lineares que as represente, como o SMILES, que é uma notação linear de organização de componentes que representa cadeias moleculares através de uma *String*.

Contudo, é muito comum o editor molecular não apresentar bloqueios, permitindo ao usuário criar as estruturas livremente, o que pode ocasionar erros, como os de valência, gerando conseqüentemente um SMILES incorreto. Isso pode gerar estruturas inconsistentes; por este motivo é imprescindível uma avaliação da sentença fornecida pelo editor. Partindo desta premissa, este projeto criou uma metodologia para verificação estrutural molecular.

A API destina-se a validação de cadeias geradas por qualquer editor de moléculas que tenha como saída o padrão SMILES absoluto, que é uma variação do SMILES. E tem como foco a criação de métodos para a detecção e apontamento de possíveis erros existentes em uma sentença SMILES.

A análise léxica garante que os lexemas utilizados estejam corretos a partir de marcas, e também separa os átomos e os classifica, segundo a tabela periódica. A análise sintática avalia a disposição dos lexemas, segundo um contexto previsto nos padrões da linguagem SMILES. Assim a API pode varrer a sentença a procura de possíveis erros estruturais. Esta prática garante que nenhuma regra gramatical da linguagem SMILES possa ter sido ferida.

O algoritmo para verificação de ligações utiliza o conceito de ligação covalente dos átomos para encontrar os erros nas estruturas, sendo necessário transformar a expressão regular do SMILES absoluto em uma expressão numérica simples, podendo assim, a partir de regras predefinidas, criar uma seqüência de cálculos em busca de algum possível erro na estrutura molecular.

Pretende-se anexar a API ao programa JChemPaint, uma ferramenta, sob licença GPL, para criação de estruturas em duas dimensões, que não possui um módulo de validação e portanto, não detecta erros estruturais nas fórmulas.

Este projeto está vinculado ao Simulador de Cálculo de Coeficiente de Partição de Moléculas Orgânicas e ambos têm apoio da FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo).

OBJETIVOS

A API para editor molecular se destina a atender um grupo de usuários da quimio-informática que utiliza a computação como ferramenta para o desenvolvimento e estudo de moléculas. Implementando uma API em linguagem orientada a objeto, mais especificamente em Java, para validação de cadeias geradas por qualquer editor de moléculas que tenha como saída o padrão CHIRASMILE.

METODOLOGIA

Foi realizado o estudo sobre o editor molecular JChemPaint para conhecimento de seus módulos e funcionalidades, além do estudo detalhado sobre a linguagem SMILES, onde padrões, regras de escrita, usabilidade e área de atuação foram definidos.

De posse destes dados, o reconhecedor léxico e sintático pode ser pesquisado, utilizando-se a teoria dos compiladores, visando à leitura da sentença como uma fita finita de caracteres. Para o reconhecedor léxico foram criados: uma expressão regular para representação da linguagem, um autômato finito determinístico para reconhecimento das cadeias e uma gramática regular para gerar palavras. O reconhecedor sintático usa a gramática livre de contexto, devido ao balanceamento sintático que a linguagem SMILES necessita, e a análise sintática descendente, para composição da árvore de reconhecimento.

Um algoritmo numérico para varredura de erros de ligação do tipo covalente foi elaborado, com capacidade de encontrar erros segundo as seguintes especificações:

a) substituir cada elemento da sentença SMILES pelo seu respectivo número de valência, obtendo uma expressão numérica;
b) a operação aritmética utilizada para resolução da expressão numérica é do tipo subtração, e depende da ligação que relaciona os elementos. Por exemplo: caso a ligação seja do tipo simples, é subtraído 1 unidade de cada elemento, caso dupla, será subtraído 2 unidades de cada elemento e assim por diante;
c) resolver primeiro a expressão que está dentro dos colchetes, salvo o caso onde haja somente um elemento;
d) resolver a expressão que está dentro dos parênteses, respeitando a ordem de dentro para fora, ou seja, o parênteses mais interno tem maior precedência;
e) em seguida, resolver o que restou na expressão numérica, obtendo um único número. Este número pode ser zero – indicando que não houve erro, ou diferente de zero – indicando a presença de erros.

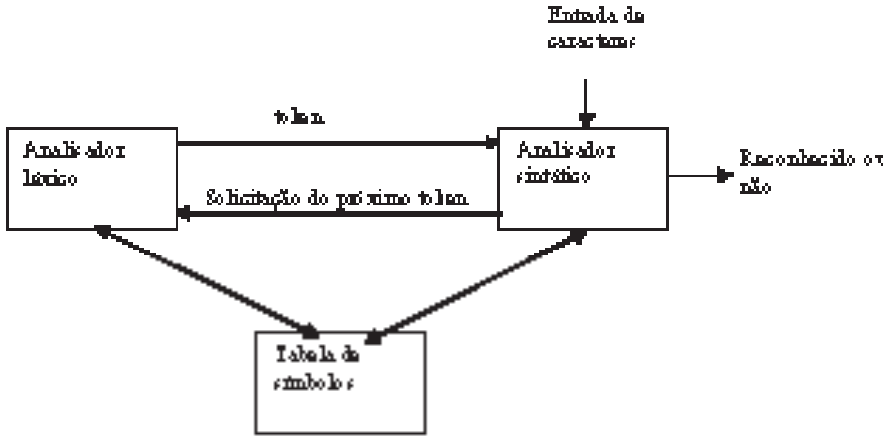
A análise de erros é obtida através das seguintes considerações:

a) o erro se encontra na posição do elemento que é diferente de zero;
b) se o número for positivo, significa ligações escassas;
c) se o número for negativo, significa excesso de ligações;
d) se 2 posições adjacentes apresentarem números idênticos, significa erro no tipo de ligação entre eles.

Em seguida, teremos mais duas etapas: uma envolvendo o JChemPaint com testes para validação da entrada e análise de dados, gerando uma saída coerente; e outra envolvendo o simulador, com testes de integração entre as duas aplicações e validação da entrada de dados, certificando que resultados inesperados não sejam gerados.

DESENVOLVIMENTO

Para construção da API, foram utilizadas técnicas de compilação e de estrutura de dados. A partir do módulo de análise léxica e sintática, é possível fazer uma varredura na sentença SMILES, em busca de possíveis erros gramaticais garantindo que os dados cheguem à função que executa os cálculos, sem erros. A seguir é mostrada uma ilustração do módulo.



O algoritmo para validação de ligações covalentes foi desenvolvido utilizando estrutura de dados e tem uma seqüência lógica de aplicação, o algoritmo transforma a sentença SMILES em uma expressão matemática onde cada elemento químico que a compõe recebe seu número de valência, o qual é consumido segundo as regras algébricas, até zerar a expressão como um todo. Caso a expressão não obtenha resultado zero, podemos afirmar que o erro se encontra na posição cujo valor é excedente como demonstrado na metodologia. O módulo de reconhecimento onde o algoritmo está alocado trabalha como no exemplo a seguir.

SMILES: O#C(C([H])([H])[H])C([H])([H])C([H])([H])[H]

O#C(C([H])([H])[H])C([H])([H])C([H])([H])[H]

2#4(4([1])([1])[1])4([1])([1])4([1])([1])[1]

2#4(4([1])([1])[1])4([1])([1])4([1])([1])[1]

2#4(2([0])([0])[1])4([1])([1])4([1])([1])[1]

2#4(1([0])([0])[0])4([1])([1])4([1])([1])[1]

2#3(0([0])([0])[0])2([0])([0])2([0])([0])[1]

-1#0(0([0])([0])[0])2([0])([0])2([0])([0])[1]

-1#-1(0([0])([0])[0])1([0])([0])2([0])([0])[1]

-1#-1(0([0])([0])[0])0([0])([0])1([0])([0])[1]

-1#-1(0([0])([0])[0])0([0])([0])0([0])([0])[0]

Como a expressão terminou com elementos que são diferentes de zero, é possível afirmar os seguintes fatos:

a) os elementos cujo resultado final foram diferentes de zero, estão incorretos;

b) o erro ocorreu por excesso de ligações nos elementos, visto que resultaram em números negativos;

c) como os dois elementos que tiveram erros são adjacentes, pode-se afirmar, que houve erro na relação de ligação existente entre os dois.

O algoritmo apesar de simples é muito eficaz e rápido, sendo capaz de oferecer várias diretivas para apontamento de possíveis erros que possam existir em uma string SMILES absoluta.

A API foi escrita em Java para manter a portabilidade com o editor molecular. Como se trata de uma API, as saídas são padrão, passando simplesmente um *array* com os objetos que proveram os cálculos, e um *flag*, com função booleana para avisar se ocorreram erros ou não. A partir destes dados é possível uma análise para conclusão se a *string* contém ou não erros e se o processo chegou ao fim de forma satisfatória.

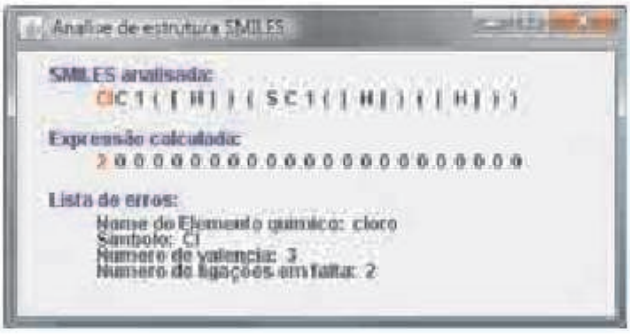
Pretendeu-se criar uma API com recursos suficientes e diversas facilidades em relação às distintas notações químicas existentes: conversões, validações, e também ser independente do editor molecular. Apesar de este trabalho ter citado como exemplo, o JChemPaint, a API aplica-se a editores cuja saídas são uma sentença SMILES absoluta, que é gerada da mesma maneira por qualquer editor, pois utiliza o algoritmo de canonização citado na teoria SMILES.

RESULTADOS

O algoritmo proposto foi aplicado com sucesso para a validação de estruturas moleculares representadas em SMILES. A seguir será apresentado um caso onde o erro está sinalizado em vermelho.

SMILES:
ClC1SC1
chiral SMILES:
ClC1([H])(SC1([H])([H]))

A molécula foi desenhada no editor molecular e gerado seu respectivo SMILES. Em seguida, a API irá apontar em vermelho os erros existentes no SMILES gerado, per mitindo ao usuário detectar a localização exata dos erros.



CONSIDERAÇÕES FINAIS

A API para validação de editor molecular foi desenvolvida com conceitos ainda pouco difundidos no Brasil. A quimio-informática é uma área pouco explorada, considerada relativamente nova, não só no Brasil, mas em todo o mundo.

A utilização do módulo reconhecedor para proceder à análise léxica e sintática eliminou, logo de início, a possibilidade de erros gramaticais invalidarem a aplicação, enquanto que o módulo de validação das ligações, proposto no algoritmo desenvolvido neste projeto, permitiu analisar possíveis anomalias nas ligações covalentes.

Portanto, a API se mostrou eficaz em detectar erros tanto na linguagem ao utilizar o analisador léxico e sintático para validar a sentença, como na estrutura molecular através do algoritmo de validação das ligações covalentes. Demonstrou boa versatilidade para análise de erros, obteve resultados satisfatórios quanto à veracidade dos dados apresentados na *string* e, por ser uma aplicação computacional, auxilia na automatização da inclusão e processamento das moléculas com maior grau de complexidade estrutural.

Há interesse em manter nesse projeto a ideologia *open source*, procurando contribuir com a comunidade cientifica, disponibilizando essa pesquisa para que outros possam aperfeiçoá-la. Tanto os códigos como os resultados obtidos serão enviados para o mantenedores do JChemPaint, nos EUA, para a possível complementação do editor, garantindo assim, um resultado confiável ao usuário final.

REFERÊNCIAS

AHO, Alfred V.; SETHI, Ravi; ULLMAN, Jeffrey D. **Compiladores:** princípios, técnicas e ferramentas. Rio de Janeiro: Guanabara Koogan, 1995.

GREVE, Georg C. F. Admirável Mundo GNU, n. 29, ago. 2001. Disponivel em: <http://bscbdsau>. Acesso em: 03 set. 2007.

JChemPaint, Introduction to. mai. 2007. Disponível em: <http://jchempaintjsbd>. Acesso em: 19 mai. 2007.

JURS, P.C.; DIXON, S.L.; EGOLF, L.M. Molecular concepts-representations. In: MANNHOLD, R.; KROOGAARD-LARSEN, P.; TIMMERMAN (Eds.). **Chemometric methods in molecular design.** Weinhein, Düsseldorf : Ham van de Waterbeemb, 1995. p. 31

LEWIS, Harry R.; PAPADIMITRIOU, Christos H. **Elementos de teoria da computação**. 2. ed. Porto Alegre: Bookman, 2000.

LOUDEN, Kenneth C. **Compiladores:** princípios e práticas. São Paulo: Pioneira Thomson Learning, 2004.

MATSUNO, Ivone Penque. **Um estudo dos processos de interferência de gramáticas regulares e livres de contexto baseado em modelos adaptativos.** 2006. 121 f. Dissertação (Mestrado em

Engenharia Elétrica) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2006.

SMILES Tokens. Disponível em: <http://www.dalkescientific.com/writings/diary/archive/2004/01/05/tokens.html> Acesso em: 3 out. 2007.

SMILES Tutorial. Disponível em: <http://www.epa.gov/medatwrk/Prods_Pubs/smiles.htm> Acesso em: 30 mar. 2007.

SOUSA, João Aires de. Quimio-informática.Conteúdo que urge ensinar. 2002. Disponivel em < http://www.dq.fct.unl.pt/staff/jas/Quimioinformatica_jas.htm>. Acesso em: 11 set. 2007.

THEORY SMILES. Disponível em: <http://www.daylight.com/dayhtml/doc/theory/theory.smiles.html> Acesso em: 28 mar. 2007.

VIEIRA, Newton J. **Introdução aos fundamentos da computação:** linguagens e máquinas. São Paulo: Pioneira Thomson Learning, 2006.

WEININGER, David. SMILES, a chemical language and information system. 1987. Disponível em: <arquivo pdf> Acesso em: 3 out. 2007.

WOLD, S.; ERIKSSON, L. Statistical validation tools, In : MANNHOLD, R.; KROOGAARD-LARSEN, P.; TIMMERMAN (Eds.). **Chemometric methods in molecular design .** Weinhein, Düsseldorf : Ham van de Waterbeemb, 1995. p. 309