Algoritmos para el cálculo eficiente de masas moleculares

*Almahir Piñango**
*Jean Paul Cholo***
*Daniel Cervera****
*Julián Mora*****
*Rubén Dorado ******

Fecha de recepción: 11 de agosto de 2014
Fecha de aprobación: 2 de octubre de 2014
Pp. 133-144

Resumen

El problema que se aborda en este trabajo es el de la representación computacional de compuestos químicos por medio de cadenas de caracteres y el cálculo de su respectiva masa molecular, de manera eficiente. En el artículo se definen dos algoritmos: el primero, llamado algoritmo base, que aunque es una solución parcial, puede utilizarse en casos en los que la velocidad sea el factor fundamental. El segundo, representa la solución, de manera general, con la que es posible representar cualquier tipo de fórmulas químicas y calcular su respectiva masa molecular asociada de manera rápida. Al final de este trabajo se prueba el desempeño de los dos algoritmos con experimentación.

Palabras clave

Algoritmos, masas moleculares, fórmulas químicas, representación computacional.

* *Pregrado en Ingeniería de Sistemas, Universidad EAN.*
** *Técnico profesional en Sistemas, Escuela Tecnológica Instituto Técnico Central La Salle. Estudiante de Pregrado en Ingeniería de Sistemas, Universidad EAN.*
*** *Estudiante del Programa de Ingeniería de Sistemas, Universidad EAN.*
**** *Estudiante del Programa de Ingeniería de Sistemas, Universidad EAN.*
***** *Magister en Ciencias de la Computación, Universidad de Tokyo. Pregrado en Ingeniería de Sistemas, Universidad Nacional de Colombia.*

Algorithms to Efficiently Calculate Molecular Mass

Abstract

This paper describes a problem which is the computational representation of chemical components through chains of characters and the calculation of its molecular mass in an efficient way. In this article, two algorithms are defined. The first one is called base algorithm which provides a partial solution applied to cases in which the speed is a key factor; the second one represents its general solution with which it is likely to represent any kind of chemical formulae and to calculate its corresponding molecular mass in a fast way. At the end of this paper, the performance of algorithms is valued through experimentation.

Key words

Algorithms, molecular mass, chemical formulae, computational representation.

Algorithmes permettant un calcul efficient des masses moléculaires

Résumé

La problématique abordée par cet article est la représentation informatique des composés chimiques réalisée grâce au calcul de la masse moléculaire respective des chaines de caractères. L'article s'attache plus particulièrement à deux algorithmes. Le premier, l'algorithme de base, est une solution partielle généralement utilisée lorsque la vitesse devient le facteur fondamental. Le deuxième algorithme est la solution par laquelle il devient possible de représenter tout type de formule chimique et de connaitre rapidement sa masse moléculaire respective. En conclusion à ce travail nous réalisons un test de performances des algorithmes.

Mots-clés

Représentation informatique, algorithmes, formules chimiques, masses moléculaires.

Algoritmos para o cálculo eficiente de massas moleculares

Resumo

O problema abordado neste trabalho é a representação computacional de compostos químicos por meio de cadeias de caracteres e o cálculo de sua respectiva massa molecular de forma eficiente. Neste artigo se definem dois algoritmos, o primeiro é chamado algoritmo base que é uma solução parcial geralmente utilizado nos casos em que a velocidade seja o fator fundamental. O segundo algoritmo representa a solução de maneira geral, com a qual é possível representar qualquer tipo de fórmula química e calcular a respectiva massa molecular associada de forma rápida. No final desse artigo se testa o desempenho dos algoritmos com experimentação.

Palavras-chave

Algoritmos, massas moleculares, fórmulas químicas, representação computacional.

1. Introducción

En muchos casos relacionados con simulaciones químicas computacionales, es necesario representar compuestos químicos de manera óptima y de esta forma, hacer operaciones, como interacciones entre las moléculas. El problema del cálculo de la masa molecular asociado a compuestos químicos consiste en hallar la densidad de un compuesto químico a partir de su fórmula o composición de moléculas que se representan computacionalmente usando cadenas de caracteres. Si bien es posible tener otros tipos de representación no lineal como estructuras de datos que representen relaciones complejas, el uso de cadenas de caracteres es necesario para reducir la cantidad de memoria utilizada y, así, maximizar el número de moléculas que se puede tener en las simulaciones.

El cálculo de la masa molecular de un compuesto químico es un proceso que se usa en diferentes escenarios. El primero de ellos, relacionado con el desarrollo de este trabajo, es el área conocida como química computacional (Cramer, 2004; Jensen, 2007); gran parte de la investigación se basa en las simulaciones químicas que pretenden modelar un proceso químico como una combinación de moléculas que interactúan entre sí dentro del computador. Por ejemplo, en Kang, Jang y Ihm (2007), se estudian varios métodos para la simulación de fluidos reactivos a nivel de partículas. En Jeschke, Park, Ewald, Fujimoto y Uhrmacher (2008), se estudian métodos paralelos para la simulación espacial de reacciones químicas y en Eberl (2003), se describen experimentos en los cuales se simulan reacciones químicas relacionadas con procesos de degradación de diferentes materiales. Los anteriores estudios son ejemplos de casos en los que la aplicación de un algoritmo, como el presentado en este estudio, podría mejorar el rendimiento de las simulaciones.

Otro tipo de aplicaciones, que no necesariamente están implicadas con simulaciones computacionales, consisten en las que se calcula la masa molecular para algún proceso industrial o químico. Por ejemplo, se han

realizado estudios para interpretar las características de los polímeros, analizando respectivamente su masa molecular como se reporta en Kilz y Held (2004). En Van Gunsteren y Berendsen (1990), se analizan las implicaciones que tienen los avances computacionales y en especial, la simulación de la dinámica de partículas en el campo de la producción química.

Por otra parte, en este artículo también se muestra la representación, a manera de cadenas de caracteres de los compuestos químicos; se presentan los dos algoritmos: el llamado algoritmo base, que es la solución usual al problema, y el algoritmo de solución general; se dan a conocer la experimentación y los resultados obtenidos y finalmente, las conclusiones.

2. Representación computacional de compuestos químicos

Existen varios modelos con los que es posible representar moléculas químicas con cadenas de caracteres. En este trabajo se usó el modelo llamado *Smiles* (Weininger, 1988). Este modelo, además de ser popular y facilitar el almacenamiento, manipulación e intercomunicación, es bastante general y permite representar moléculas complejas lo que ha permitido numerosas aplicaciones. Para este estudio, se usó una simplificación de este esquema, el cual se explica de manera formal.

Dado un conjunto de “n” elementos químicos Q, donde cada elemento está representado por una secuencia de 1 o más caracteres, una molécula “M” es una secuencia de elementos químicos básicos donde al final de cada elemento es posible agregar un número que representa repetición de un número de veces. Por ejemplo, dado el conjunto de elementos químicos básicos Q= {“C”,“O”,“H”,“N”}, algunos compuestos (moléculas) que podrían aparecer son: “CO_2”, “H_2O”, “$H_3$4NH”, “$H_2N_2H_6$”, “H35H”. Adicionalmente a esta representación, se tiene un mapa donde cada

elemento químico tiene un peso asociado; de esta manera, es posible calcular el peso que tiene asociado un compuesto químico representado por una cadena de caracteres.

El problema por resolver es entonces, diseñar un algoritmo que realice el cálculo del peso de la molécula que se está representando con la cadena de caracteres. La solución base, dadas ciertas condiciones como que cada compuesto está representado por un único carácter, es un típico algoritmo de recorrido de cadenas con un cálculo asociado. Este algoritmo, llamado aquí solución base, es explicados en la siguiente subsección. Sin embargo, la solución base no considera casos en los que las cadenas “N”, “NN”, “NNN” podrían ser elementos, o no considera elementos cuya representación involucre más de un carácter “Cl”, “Mg” o “Ca”. La solución para el problema general se explica después con el algoritmo de tipo *backtracking* recursivo.

3. Algoritmos

Esta sección muestra los dos algoritmos propuestos. La primera subsección describe un algoritmo llamado algoritmo base, que es una solución parcial del problema y solo puede ser usado en problemas que involucren elementos de un solo caracter, pero que brindan un desempeño superior en tiempo. La segunda subsección presenta un algoritmo que calcula el peso de moléculas con elementos químicos básicos que estén representados con varios caracteres.

3.1 Algoritmo 1: solución base

En la solución base se dan por hecho algunos supuestos, siendo el más fuerte, el que cada uno de los elementos está formado por un único carácter, lo que implica que no hay elementos formados por otros elementos. El código 1 presenta el pseudocódigo del algoritmo base que

calcula la masa molecular de una de cadena de “m” caracteres con esta representación. El recorrido de la cadena tarda un tiempo de Θ(m) ya que recorre toda la cadena de caracteres. Cada vez que se procesa un carácter (líneas 7 a 15), se revisa si es este es un número y, de ser así, se acumula para procesar el caso en el que se trate de un número de varios caracteres. Es importante notar que este proceso de acumulación es lineal y por lo tanto, no cambia el rendimiento del algoritmo. En el caso de que se encuentre una letra, se busca en el arreglo de elementos químicos (línea 11) para obtener su valor asociado. La búsqueda en un arreglo (o lista) es lineal y tomando como “n” el número de elementos, la búsqueda se realiza en O(n), por lo que el rendimiento del algoritmo está regido por Θ(m)*O(n)=O(mn).

```
 1 solucion_base (char^n:s)
 2
 3   resp ← 0
 4   num_acum ← ""
 5   elem_acum ← ""
 6
 7   para cada caracter s[i]
 8      si esNumero (s[i])
 9         num_acum ← concatenar(num_acum, s[i])
10      si no
11         resp ← resp +  buscarMasa(elem_acum)*num_acum
12         num_acum ← ""
13         elem_acum ← s[i]
14      fin si
15   fin para
16
17   retorne resp
18 fin procedimiento
```

Es posible mejorar este rendimiento evitando una búsqueda al guardar temporalmente todos los compuestos y asociarlos con el número de veces que aparecen sin buscar su peso asociado o usando una estructura de guardado como un TRIE. Con estas mejoras, es posible generar que el algoritmo termine en un peor tiempo de O(n). Sin embargo, como se explicó anteriormente, este algoritmo no resuelve el caso general en el que un compuesto puede estar formado por varios caracteres. A continuación se propone un algoritmo que encuentra la solución general.

3.2 Algoritmo 2: solución general

```
 1 algoritmo2(char^n:str, num: indx)
 2
 3   i ← indx
 4   acum ← ""
 5   cont ← true
 6   mult ← ""
 7
 8   mientras cont y i < largo(str)
 9      acum ← acum+str[i]
10
11      mientras isNumber(str[i+1])
12         cont ← false
13         mult ← mult+ str[i+1]
14         i ← i + 1
15      fin mientras
16
17      si existe(acum)
18         si i=largo(str)
19            returne val(acum)*mult
20         sino
21            tmp = algoritmo2(str, i)
22            si tmp ≠ -1
23               returne tmp + buscarMasa(acum)*mult
24            fin si
25         fin si
26      fin si
26   fin mientras
27
28   returne -1
29 fin procedimiento
```

Esta apartado presenta un algoritmo que calcula la masa molecular de compuestos que contengan cualquier tipo de elementos, incluyendo definiciones no convencionales como “CL”,”CA”,“NN”, “ANON”, etc. El algoritmo es de tipo backtracking avaro;, es decir, que por medio de recursividad realiza una búsqueda exhaustiva de todas las posibles combinaciones, devolviendo la mejor solución que encuentreo primero.

El código 2 muestra el pseudocódigo del algoritmo propuesto que encuentra la solución para cualquier compuesto, según la representación discutida anteriormente. El algoritmo trata de buscar subsecuencias de compuestos leyendo caracteres desde un índice que se provee por medio de un parámetro obligatorio “indx línea 1”. De la misma manera que con el algoritmo 1, se utiliza un acumulador para diferenciar caracteres numéricos que representan repeticiones de elementos, agregándolos a un acumulador y luego buscándolos en la tabla de elementos. Si el componente acumulado no se encuentra (línea 17), no entra dentro de la estructura condicional y retorna un valor que indica que esa subsecuencia no es válida en la línea 28 y continúa con el siguiente, llamado recursivo, si existe. El elemento existe en la tabla de elementos válidos. En la línea 14 entra al condicional haciendo un llamado recursivo en la línea 21 que

termina de probar el resto del compuesto con el restolos demás de la secuencia de caracteres. De esta manera, el segundo parámetro es de gran relevancia ya que indica el punto desde el que se va a probar la subcadena en cada llamado. También es importante resaltar que el caso base del algoritmo se puede ver en la línea 21, es decir, cuando el parámetro *indx* es el tamaño del compuesto o, en otras palabras, se llega al final de la cadena de caracteres.

El análisis del rendimiento del algoritmo es mucho más complejo que el mostrado en el algoritmo 1, principalmente por su naturaleza recursiva. Para calcularla se hará uso del teorema maestro, método probado para calcular la complejidad de algoritmos recursivos (Cormen, Leiserson, Rivest, & Stein, 2009). El teorema maestro se usa calculando las relaciones de recurrencia, en este caso T(n,m)=T(n-1,m)+O(n) que es la misma relación de algoritmos conocidos como el insertion sort o selection sort, y cuya relación de recurrencia se resuelve dando como peor caso T(n,m)=O(n^2 m), resultado similar al peor caso encontrado en el algoritmo 1. Sin embargo, este peor caso es, en realidad, un caso particular con condiciones muy especiales. Si el algoritmo se ejecuta con las mismas entradas que las del algoritmo 1, este solo hace un llamado por cada letra y en el caso de que el compuesto químico contenga números, funciona en un peor tiempo de O(mn); o sea, exactamente igual que el caso base.

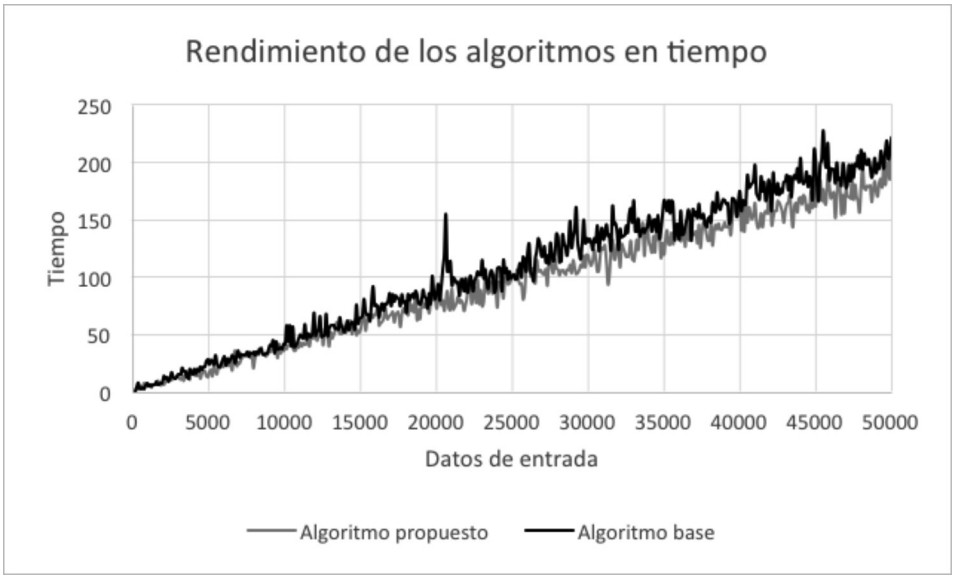
Es posible, así mismo, que con el algoritmo 1, se mejore el rendimiento al usar una estructura óptima para la búsqueda de cadenas de caracteres como un TRIE o contar primero los compuestos químicos y luego realizar el cálculo. La primera opción es de fácil implementación, al igual que con el algoritmo 1. No obstante, la segunda estrategia es un poco más complicada e implica hacer cambios drásticos en el algoritmo.

4. Experimentos y resultados

Para la experimentación de los dos algoritmos se usó un programa que genera moléculas probabilísticamente. Se utilizó el conjunto de las letras del abecedario en mayúsculas como elementos químicos; es decir, 24 elementos químicos de un solo carácter. Para comparar el rendimiento de los dos algoritmos experimentalmente, se creó un conjunto de datos de prueba de 50.000 fórmulas de compuestos químicos con su respectivo peso calculado.

Los dos algoritmos se ejecutaron guardando los tiempos para cada cantidad de datos de entrada, usando validación cruzada; en otras palabras, seleccionando un número de datos al azar y seleccionando , así como el peor tiempo para obtener los peores casos. Se muestran los resultados del experimento donde el eje “x” representa la cantidad de datos de entrada para el algoritmo, y en el eje “y” se reporta el peor tiempo obtenido para esa entrada en milisegundos (Figura 1). En gris se muestra el resultado para el algoritmo propuesto y en negro, los tiempos del algoritmo base.

Figura 1. Comparación de los dos algoritmos discutidos



Se puede llegar a dos conclusiones: la primera, que los tiempos del algoritmo base son ligeramente mayores;. por ejemplo, mientras el algoritmo base procesó los 50.000 datos en 222 milisegundos, el algoritmo 2 lo hizo en 198 milisegundos. El segundo resultado conclusivo y más importante, radica en que los resultados confirman que se trata de un tiempo lineal. Como el tamaño de los elementos básicos es pequeño (el número de letras), el número de elementos químicos “m” en el peor tiempo analítico O(mn) no es significativo y los resultados aparentan ser lineales. Para obtener la gráfica cuadrática del verdadero tiempo de las dos se tendría que usar un número de elementos químicos significativo.

5. Conclusiones

Este artículo presenta dos algoritmos que solucionan el problema del cálculo de masa molecular de compuestos químicos a nivel computacional. Los análisis que se hacen sobre los dos algoritmos demuestran que el propuesto en este artículo, reporta tiempos de ejecución óptimos. A pesar de que el algoritmo propuesto es de tipo *backtracking*, reporta tiempos similares a un algoritmo lineal (algoritmo 1).

También es importante resaltar, que es posible mejorar el rendimiento de ambos algoritmos usando una estructura óptima para la búsqueda de cadenas de caracteres como un TRIE o contar primero los compuestos químicos y luego calcular el peso molecular.

6. Referencias bibliográficas

Cormen, T. H., Leiserson, C. E., Rivest, R., & Stein, L. (2009). Introduction to Algorithms. MIT Press.

Cramer, C. J. (2004). Essentials of Computational Chemistry: Theories and Models. England: Wiley.

Eberl, H. J. (2003). Simulation of chemical reaction fronts in anaerobic digestion of solid waste. Proceedings of the 2003 international conference on Computational science and its applications: PartI (ICCSA'03) (págs. 503-512). Berlin: Springer-Verlag.

Jensen, F. (2007). Introduction to Computational Chemistry. England: Wiley.

Jeschke, M., Park, A., Ewald, R., Fujimoto, R., & Uhrmacher, A. M. (2008). Parallel and Distributed Spatial Simulation of Chemical Reactions. Proceedings of the 22nd Workshop on Principles of Advanced and Distributed Simulation (PADS '08) (págs. 51-59). Washington: IEEE Computer Society.

Kang, B., Jang, Y., & Ihm, I. (2007). Animation of chemically reactive fluids using a hybrid simulation method. Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA '07) (págs. 199-208). New York: ACM.

Kilz, P., & Held, D. (2004). Structure Elucidation with Molar Mass Sensitive Detectors. LC-GC Europe, 1739.

Van Gunsteren, W. F., & Berendsen, H. J. (1990). Computer Simulation of Molecular Dynamics: Methodology, Applications, and Perspectives in Chemistry. En Angewandte Chemie International Edition in English (págs. 992–1023). Wiley.

Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. Journal of Chemical Information & Computer Sciences, 28(1), 31 - 3