화학 데이타베이스에서 부분구조 검색을 위한 인덱스 구조

(An Index Structure for Substructure Searching In Chemical Databases)

이 환 구 † 차 재 혁 ††
(Hwangu Lee) (Jaehyuk Cha)

요 약 약물의 화학적 구조와 그 약물의 약리작용간의 연관성은, 'Medicinal Chemistry' 분야에서 활발히 연구된다. 이는 화학구조를 기반으로 하여 신약을 설계하려는 시도로서, 약학자는 신약 개발 시 만들고자 하는 약물과 비슷한 화학구조를 가지고 있는 기존 약물들에는 어떠한 것들이 있는지 조사하며, 특정 화학구조가 어떤 약물들에서 나타나는지 신속히 검색하기를 원한다.

이처럼 어떤 화학구조에서, 특정한 부분구조가 존재하는지를 검사하는 것을 부분구조검색(Substructure Searching)이라 하며, 이는 그래프 이론에서 NP-complete인 동형성 판정(Subgraph Isomorphism) 문제로 귀결된다. 검색 시간을 단축시키고자 여러 다른 접근방법들이 연구되었는데, 1990년대에는 구조에 대한 인덱스를 미리 만들어 RDBMS에 저장한 후, 검색시 이를 이용하여 성능을 높이는 방법으로 미국 특허를 획득한 RS3 시스템(http://www.acelrys.com/rs3)이 현재 상용화되어 쓰이고 있다.

본 논문에서는 RS3 시스템의 문제점을 규명하고, 이의 개선방안으로서 새로운 인덱스를 제안한다. RS3 시스템은 각 원자를 중심으로 다른 원자와의 구조를 문자열로 표현하고, 부분구조검색 쿼리를 부분문자열 검색을 실행함으로써 수행하는데, 이의 화학구조를 기술하는 인덱스에는 동일 원자, 동일 결합에 대한 정렬이 불가능하여 재현율(Recall)과 정도(Precision)가 낮다. 이를 개선하기 위하여 본 논문에서는 2차원의 화학구조를 나누어 1차원의 구조 단편으로 만들고 이를 문자열로 기술하는 방안을 제시하며 구체적인 방법으로 한 원자를 중심으로 최소비용신장트리를 구성한 다음 레벨별로 경로를 나누어 기술하는 방안을 제안하며, 이와 같은 방법의 새로운 인덱스로 재현율과 정도가 급격히 향상됨을 보인다.

키워드 : 부분구조검색, 동형성 판정, RS3 시스템

Abstract The relationship between chemical structures and biological activities is researched briskly in the area of 'Medicinal Chemistry'. At the base of these structure-based drug design tries, medicinal chemists search the existing drugs of similar chemical structure to target drug for the development of a new drug. Therefore, it is such necessary that an automatic system selects drug files that have a set of chemical moieties matching a user-defined query moiety.

Substructure searching is the process of identifying a set of chemical moieties that match a specific query moiety. Testing for substructure searching was developed in the late 1950s. In graph theoretical terms, this problem corresponds to determining which graphs in a set are subgraph isomorphic to a specified query graph. Testing for subgraph isomorphism has been proved, in the general case, to be an NP-complete problem. For the purpose of overcoming this difficulty, there were computational approaches. On the 1990s, a US patent has been granted on an atom-centered indexing scheme, used by the RS3 system; this has the virtue that the indexes generated can be searched by direct text comparison. This system is commercially used(http://www.acelrys.com/rs3).

We define the RS3 system's drawback and present a new indexing scheme. The RS3 system treats substructure searching with substring matching by means of expressing chemical structure aspredefined strings. However, it has insufficient 'recall' and 'precision' because it is impossible to

· 본 연구는 과학기술부 프로테오믹스 이용기술 개발사업 (과제번호: M102KM010014-04K1301-01411)의 지원을 받았음
† 비 회 원 : 한양대학교 소프트웨어공학
prodog@ihanyang.ac.kr
†† 종신회원 : 한양대학교 정보통신학부 교수
chajh@hanyang.ac.kr
논문접수 : 2003년 12월 22일
심사완료 : 2004년 11월 29일

index structures uniquely for same atom and same bond. To resolve this problem, we make the minimum-cost-spanning tree for one centered atom and describe a structure with paths per levels. Expressing 2D chemical structure into 1D a string has limit. Therefore, we break 2D chemical structure into 1D structure fragments. We present in this paper a new index technique to improve recall and precision surprisingly.

Key words : Substructure Searching, Subgraph Isomorphism, RS3 system

1. 서 론

1.1 연구 배경

새로운 약을 개발하려는 여러 가지 방법 중에, 'Medicinal Chemistry' 분야는 약물의 화학적 구조와 그 약물의 약리작용간의 연관성에 대해 연구한다[1]. 즉 화학구조를 기반으로 하여 신약을 설계하려는 시도이다. 이러한 시도의 기초가 되는 연구로서, 약학자는 신약 개발 시, 만들고자 하는 약물과 비슷한 화학구조를 가지고 있는 기존 약물들에는 어떠한 것들이 있는지 조사한다. 다시 말하면, 연구자나 개발자들은 특정 화학구조가 어떤 약물들에서 나타나는지 신속히 검색하기를 원하고 있다. 따라서 수많은 약물들에서, 특정 화학구조를 부분구조로 가지는 약물들을 자동으로 빠르게 찾아내는 것이 필요하다.

이처럼 어떤 화학구조에서 특정한 부분구조가 존재하는지를 검사하는 것을 부분구조검색(Substructure Searching)이라 한다. 부분구조검색은 1950년대부터 많은 사람들이 연구하였다. 부분구조검색은 그래프 이론에서 동형성 판정(Subgraph Isomorphism) 문제로 귀결되고, 이는 NP-complete 문제이다. 이를 극복하고자 다른 접근방법들이 연구되었다. 예를 들면 좀 더 빠른 컴퓨터를 사용하거나, 하드웨어 병렬화 기법을 사용하는 방법이 그것이다. 그리고 검색 후보가 되지 않는 원자들은 검색 대상에서 제외시키는 알고리즘이나 발견적 방법(Heuristics)이 있다.

이러한 방법들은 여전히 많은 시간을 요구했기 때문에, 이러한 알고리즘의 개선과는 별도로 연구된, 시간이 많이 소모되는 연산은 미리 계산하는 스크리닝(Screening) 기법이 개발되었다. 구조를 대표하는 키(Key)를 개발하여 이를 미리 계산하여 저장함으로써, 검색시 속도를 개선시킨다. 1980년대 들어 하드디스크 가격이 낮아짐과 맞물려, 각 원자를 중심으로 구조를 상세히 기술하는 대용량의 키를 개발하는 방법이 등장했다. 1990년대에 원자 중심 구조 기술에 대한 인덱스(Index)를 문자열 화하여 RDBMS에 저장하여 놓고, 검색시 문자열 검색을 이용하여 검색 속도를 향상시킨 RS3 시스템(http://www.accelrys.com/rs3)이 미국 특허를 받았다. 그리고 현재 상용화되어 쓰이고 있다.

하지만 RS3 시스템은 화학구조를 기술하는 인덱스에 치명적인 결함이 있어, 재현율(Recall)과 정도(Precision)가 매우 낮다.

1.2 연구 내용

RS3 시스템의 장점은 어떤 원자를 중심으로 다른 원자와의 구조를 문자열로 기술한 인덱스를 만들어 이를 검색시 활용함으로써, 부분구조검색을 부분문자열검색으로 변환시켜 NP-complete인 본 문제를 O(n)의 시간복잡도로 줄인 점이다. 즉 각각의 원자마다 다른 원자와의 결합관계를 문자열로 표현한다. 검색시에는 쿼리(query)할 화학구조도 각각의 원자마다 구조를 문자열로 기술하되, 와일드카드(%)를 적절히 삽입하여 이들 문자열로 문자열부분검색을 수행한다. 하지만 RS3 시스템은 구조 기술 법에서 치명적인 결함이 존재하여, 재현율과 정도가 낮다.

본 논문에서는 RS3 시스템의 장점은 그대로 살리면서, 인덱스의 결함은 극복한 새로운 인덱스를 제시한다.

2. 관련 연구

본 장에서는 부분구조검색에 대한 기존 연구에 관하여 기술한다.

2.1 그래프 이론

그래프 이론 측면에서 보면 화학구조에서 원자는 노드(node)로 결합(bond)은 에지(edge)로 간주된다.

화학부분구조검색은 동형성 판정(Graph Isomorphism)으로 귀결된다. 동형성 판정(Graph Isomorphism)은 NP-complete 문제이다[2].

2.2 다른 접근 방법들

NP-complete인 이 문제에 대해 여러 가지 다른 접근방법들이 있다. 이를 요약하면 아래와 같다.

1. 좀 더 빠른 컴퓨터를 사용하거나, 하드웨어 병렬화 기법을 사용한다.
2. 기억장치 사용효율(Space Utilization), 디스크 접근(Disk Access), 수행 시간(Execution Time)을 고려한 알고리즘을 개발한다.
3. 타겟(target)이 되는 후보구조를 줄이는 알고리즘이나 발견적 방법(Heuristics)를 사용한다.
4. 각각의 쿼리/타겟 비교에서 매핑(mapping)해야 할 노드나 에지의 수를 줄이거나 정렬하는 기법을 개발한다.

5. 일반화된 구조 표기법에 따라 타겟들을 미리 계산하여 저장한다.

첫 번째 방법은 CAS(Chemical Abstracts Service)[3], Daylight Chemical Information System[4], Synopsys[5] 등에서 사용한다.

2.3 역추적(Backtracking) 알고리즘

2.2의 세 번째와 네 번째에 해당하는 것이 역추적(Backtracking) 알고리즘이다. 1957년에 Ray와 Kirsch에 의해 개발된 이 알고리즘[6]의 개요는 현시점에서 더 이상 일치하는 노드가 없으면 바로 전 노드로 거슬러 올라가서 다른 노드를 골라 비교해 보는 것이다.

이 알고리즘은 계속 정교해졌는데, 이를 세 그룹으로 나누면 아래와 같다.

1. 타겟에서 후보가 되지 않는 원자수를 계속해서 줄여 나간다.
2. 쿼리와 타겟 원자의 우선순위를 매긴다.
3. 역추적(Backtracking) 알고리즘을 종료할 조건을 정한다.

2.3.1 분할(Partitioning)과 완화(Relaxation)

분할(Partitioning)이란 각 원자의 연결지수(connectivity), 결합 차수(bond order) 등과 같은 부분적인 특성을 질의 구조의 각 원자와 비교함으로써, 후보가 되는 원자를 걸러내는 방법이다.

완화(Relaxation)이란 분할(Partitioning)을 반복적으로 이웃 원자들에게 적용시켜 나가는 것이다.

2.3.2 Ullmann 알고리즘

동형성 판정(Subgraph Isomorphism) 문제의 대표적인 알고리즘이 Ullmann 알고리즘[7]이다. 이 알고리즘의 개요는 타겟 노드 T_i에 매핑되는 쿼리 노드 Q_i가 이웃 노드 Q_j를 가지고 있다면, T_i도 Q_j에 매핑되는 T_j를 가지고 있어야 한다는 것이다. 이를 반복적으로 매핑하지 않은 원자들에게 적용함으로써 비교할 후보 원자수를 줄이는 것이다.

2.4 스크리닝

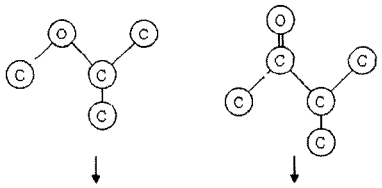
원자와 원자 간의 비교 알고리즘은 여전히 많은 시간을 요구한다. 알고리즘을 개선하는 것과 병렬적으로, 비교해야할 구조들을 미리 걸러내어 비교 연산 수를 줄이는 스크리닝 시스템이 개발되었다. 구조를 대표하는 키를 정의하고 이를 미리 계산하여 데이타베이스에 저장하고, 검색시 질의 구조와 타겟 구조의 키를 비교함으로써, 후보가 되는 타겟 구조들을 가려내게 된다.

2.4.1 키의 개발

키를 개발하는 것이 스크리닝 시스템을 설계하는데 있어 가장 중요하다. 1970년대 초에 Lynch et al.에 의해 자주 발견되는 것은 식별력(discrimination)이 낮지만 빈번하게 쓰이고, 잘 발견되지 않는 것들은 식별력은 높지만 자주 쓰이지 않는다는 것을 발견했다[8]. 이 사실을 근거로 BASIS fragment dictionary[9]를 만들게 되었고, 이는 후에 STN International의on-line substructure search system[3]에 쓰이게 되었다.

2.4.2 키의 저장 형태

대부분 키의 구현은 비트 스트링(bit string)으로 한다. 그 한 형태가 그림 1로서, 데이타베이스에 저장된 화학 구조 파일이 미리 정의된 구조를 가지고 있는지를 한 비트로 표시한다. 데이타베이스의 n번째 항목이 m번째 구조를 가지고 있는지의 여부는 n번째 비트 스트링의 m번째 비트가 표시한다. 쿼리 구조도 같은 방법으로 비트 스트링을 만들고 이를 비교함으로써 쿼리 구조를 부분 구조로 가지는 데이타베이스의 항목을 찾게 되는 것이다. 예를 들어 그림 1의 좌측 구조는 C, O, C-O, C-C-O, C-C(-C)-O를 가지고 있으므로 데이타베이스상에 2번째 화학 구조의 부분 구조이게 된다. 마찬가지로 우측 구조는 7번째 항목의 부분 구조이다.



	1	2	3	4	5	6	7	8	9	10
C	1	1	1	1	1	1	1	1	0	1
O	0	1	0	1	1	1	1	1	0	1
C-O	0	1	0	1	0	1	0	1	0	1
C=O	0	0	0	1	0	0	1	0	0	0
C-C-O	0	1	0	0	0	0	0	0	0	1
C-C=O	0	0	0	1	0	0	1	0	0	0
C-C(-C)-O	0	1	0	0	0	0	0	0	0	1
C-C(-C)=O	0	0	0	1	0	0	1	0	0	0

그림 1 비트 스트링으로 구현된 스크리닝 시스템의 데이타베이스

검색시 이 비트 스트링에 대한 AND 연산으로 원자간 비교를 수행할 후보 구조들을 산출하게 된다.

2.4.3 키의 계층적 분류

키가 부분적으로 같은 것끼리는 묶어 계층적으로 저장해 놓아 성능을 향상시키는 기법으로 ORAC[10], CIS(National Institute of Health)[11], DARC[12] 등에 쓰이고 있다.

2.4.4 축약(Reduced) 그래프와 하이퍼그래프(Hypergraphs)

축약(Reduced) 그래프란 그래프의 일부분을 하나의 노드로 표현한 그래프이다. 이와 같이 함으로써 비교할 노드와 에지 수를 줄이게 된다.

하이퍼그래프(Hypergraph)란 여러 개의 그래프를 가지고 일치되는 부분을 중첩시켜 하나의 그래프로 만든 그래프이다. 이와 같이 하면 저장 공간도 줄어들게 되고 여러 개의 그래프를 검색할 필요 없이 하나의 큰 그래프만 검색하면 된다. 하이퍼그래프(Hypergraph)를 만드는 것은 최대 공통 부그래프(Maximal Common Subgraph)를 구하는 것으로 NP-complete 문제이지만 이는 미리 연산하여 저장한다.

2.4.5 원자 중심 인덱스(Atom-centered Indexes)

1980년대 들어 미리 정의된 스크린(screen)을 사용하지 않고, 원자를 중심으로 알고리즘 적으로 인덱스를 생성하는 두개의 시스템이 개발되었는데, 바로 HTSS (Hierarchical Tree Substructure Search)[13,14]와 S4 시스템[15]이다.

1990년대에는 이러한 인덱스를 직접 문자열 검색을 할 수 있게 개발된 RS3 시스템[16]이 미국 특허를 얻었다[17].

2.4.5.1 HTSS

그림 2처럼 정해진 규칙에 따라 원자들을 계층적으로 구분지어 저장한다. 쿼리도 마찬가지로 분류한 후 매칭하게 된다.

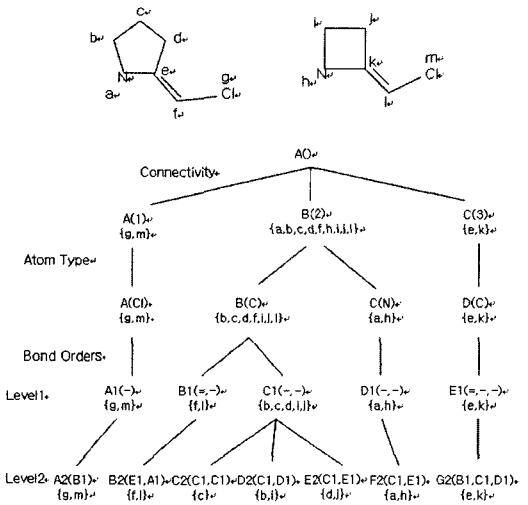


그림 2 HTSS

2.4.5.2 S4 시스템

S4 시스템은 각각의 원자별로 다른 원자들까지의 경로(path)를 매우 간결한 코드(code)로 생성한다. 이 코드를 검색하는 것만으로도, 쿼리를 완전히 수행할 수 있다면, 검색 알고리즘은 원자별 매핑을 시도하지 않아도 된다. 이러한 코드들은 계층적으로 저장되는데, 압축되고 정렬됨으로써 무작위한 디스크(disk) 액세스(access)를 줄인다. 만일 코드가 쿼리를 수행하는데 부족하다면, 원자별 매핑을 수행한다.

3. RS3 시스템

본 장에서는 본 논문의 모태가 되는 RS3 시스템에 관해 기술한다.

3.1 RS3 시스템의 장점

RS3 시스템의 장점은 원자 중심의 인덱스를 하되, 다른 원자와의 결합관계를 문자열로 표현하고, 쿼리도 와일드카드(%)를 적절히 포함한 문자열로 표현하여, 부분구조검색을 부분문자열검색으로 변환시킨 점이다. 이렇게 함으로써 NP-complete인 부분구조검색 문제를 O(n)의 시간복잡도로 줄일 수 있다.

부분문자열검색으로 변환시키기 위하여 우선, 표 1처럼 이웃원자와의 결합관계를 하나의 문자로 정의한다. 예를 들어 Br원자와의 1차결합은 'b' 문자로 표시하고, C원자와의 2차결합은 'd' 문자로 표시한다.

표 1 RS3 시스템에서 원자간 결합 관계의 축약

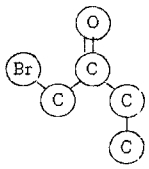
- Br	b
- C	c
= C	d
= O	e
wildcard	%

그 후, 그림 3처럼 각 원자별로 '.'으로 단계를 구분하여 결합구조를 문자열 화한다. 이때 문자간의 순위는 알파벳 순서를 따르며, 순서대로 기술되지 않은 부분을 '.'으로 나누며 기술한다. 예를 들어, 아래 그림의 Br원자에 대한 나머지 원자들의 결합 구조를 문자열화하면 다음과 같다.

1. 먼저 Br원자에 한 개의 C원자와의 1차결합이 존재하므로 "Br c"으로 쓰고, '.'으로 단계가 구분됨을 표시하여 "Br c ."으로 기술한다.
2. 다음에 이 C원자에 기술된 Br원자를 제외한 다른 C원자가 1차결합으로 붙어 있으므로 "Br c . c ."으로 문자열을 확장한다.
3. 다음으로 마지막으로 기술된 C원자에 C원자와의 1차결합, O원자와의 2차결합이 존재한다. 이는 각각 'c'와 'e'로 표시되고, 알파벳 순서에 따라 "c e"로 문자열에 붙게 된다. 그리고 단계를 나누는 '.'문자까지 기술하면 현재까지의 문자열은 "Br c . c . c e ."이 된다.
4. 현재 문자열은 "Br c . c . c e ."이고, 여기에서 "c e"에 해당하는 C원자와 O원자 중 다음으로 기술할 순서는 알파벳 순서대로 C원자이다("c e"에서 'c'). 그 C원자에는 또 다른 C원자가 1차 결합되어 있으므

로 문자열은 "Br c . c . c e . c ."가 된다.

5. 다음으로 기술할 O원자에는 기술되지 않은 이웃원자가 존재하지 않으므로 단지 '.'만 붙여준다. 따라서 문자열은 "Br c . c . c e . c . ."가 된다.
6. 마지막으로 기술된 원자("Br c . c . c e . c . ."에서 마지막 'c'에 해당하는 C원자)에도 기술되지 않은 이웃원자가 존재하지 않으므로 단지 '.'만 붙여주어 문자열은 "Br c . c . c e . c . . ."가 된다.
7. 이제 모든 원자에 대한 기술이 끝났으므로, 최종 구조 기술 문자열은 "Br c . c . c e . c . . ."가 된다.



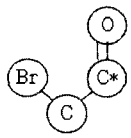
1. Br c . c . ce . c . . .
2. C bc . . ce . c . . .
3. C c c e . b . c . . . .
4. O d . c c . b . c . . .
5. C c c . . ce . b . . .
6. C c . c . ce . b . . .

그림 3 RS3 시스템에서 DB저장구조와 문자열

부분구조를 쿼리하는 경우는 그림 4와 같은데, 여기서 결합될 수 있는 부분(쿼리하는 부분구조가 전체 화학구조에 붙을 수 있는 곳)을 '*'으로 지정해 주어야 한다. '*'가 지정된 원자는 부분구조를 찾아야하는 곳이 되고 이는 문자열 검색에서 부분문자열검색을 하여야 하는 곳이 된다. 즉, 와일드카드(%)를 문자열안에 적절히 조합해야 하는 곳이다. 쿼리구조에 대한 문자열 기술도 지금까지와 같은 방법대로 하며, 와일드카드 배합법을 Br 원자를 예로 들어 설명하면 다음과 같다.

1. 먼저 Br원자에 한 개의 C원자와의 1차결합이 존재하므로 "Br c"으로 쓰고, '.'으로 단계가 구분됨을 표시하여 "Br c ."으로 기술한다.
2. 다음에 이 C원자에 기술된 Br원자를 제외한 다른 C원자가 1차결합으로 붙어 있으므로 "Br c . c ."으로 문자열을 확장한다.
3. 마지막으로 기술된 C원자에는 O원자와의 2차결합이 존재하며, '*'로 지정되었다. 이는 여기에는 O원자 말고 다른 원자가 결합되어도 된다는 것을 의미한다. 따라서 와일드카드(%)를 O원자와의 결합을 기술하는 문자 'e'에 앞뒤로 붙여주어 부분문자열 검색에 쓰일 수 있도록 한다. 따라서 문자열은 "Br c . c . % e % ."이 된다.
4. 현재까지의 문자열에서 와일드카드(%)가 나타나게 되면, 다음으로 다시 와일드카드(%)를 덧붙이고 문자열 생성을 종료한다. 따라서 최종 문자열은 "Br c . c . % e % . %"가 된다.

위의 쿼리 문자열을 가지고 DB에 저장된 문자열들과 문자열 검색을 수행하게 되면, 후보가 되는 화학구조파



1. Br c . c . % e % . %
2. C b c . . % e % . %
3. C % c % e % . %
4. O d . % c % . %

그림 4 RS3 시스템에서 쿼리부분구조와 문자열

일들을 추릴 수 있다. 그 후 ABAM(Atom-by-Atom Matching)을 수행하여 정확한 결과구조를 출력한다.

RS3 시스템은 현재 Accelrys사에서 상용화되었다[16].

3.2 RS3 시스템의 문제점

RS3 시스템의 가장 큰 문제점은 '*'로 지정한 곳(부분구조가 연결될 수 있는 곳)이 많아짐에 따라 정도가 급격히 낮아진다는 점이다. 그림 4의 3번째 문자열 "C % c % e % . %"처럼 '*'이 지정된 곳을 만나면 와일드카드(%)가 삽입되고("C % c % e %"), 다음으로는 ".%"으로밖에 쓰여질 수 없어 더 이상의 구조기술이 어렵게 된다. 즉, '*'를 만나게 되면 그 이후의 구조 기술은 불가능하게 되는 것이다. 쿼리구조는 Br원자와의 1차결합이 존재하나, 이 문자열은 다음으로 어떠한 원자나 결합이 존재하여도, 다시 말하면 어떠한 구조가 와도 허용하게 되므로, 검색시 정도를 떨어뜨린다.

또 다른 문제점은, 동일원자의 동일결합에 대해 우선순위를 정할 수 없다는 점이다. RS3 시스템의 중요한 아이디어는, 구조 기술시 우선순위를 알파벳 순서로 한다는 점이었다. 구조 기술시, 어떤 결합부터 기술할지에 대해 순서를 부여할 수 있음으로 해서, 어떤 하나의 구조에 대해 하나의 문자열을 생성할 수 있는 것이다. 예를 들어 그림 3의 2번째 문자열인 "C b c . . c e . c . . ."은 C원자에 Br원자와의 1차결합, C원자와의 1차결합이 있어, 이를 나타내는 문자 'b', 'c'를 알파벳 순서대로 나열한 것이다. 즉 이 구조에 대한 문자열은 이 문자열 하나밖에 존재할 수 없는 것이다. 하지만 3번째 문자열 "C c c e . b . c . . . ."에서처럼 같은 'c', 'c' 문자에 대해서는 우선순위를 정할 수 없기 때문에 이 구조는 "C c c e . c . b . . . ."이라고도 기술될 수 있다. 이처럼 이후의 구조를 어느 것부터 기술하느냐 하는 것이 명확하지 않음이, RS3 시스템의 재현율을 낮게 하는 이유가 된다.

4. 개선된 RS3 시스템

본 장에서는 RS3 시스템을 개선시킨 방안을 제시한다.

4.1 RS3 시스템의 문제점 해결 방안

RS3 시스템의 정도와 재현율을 높이기 위해서는 구조 기술법을 바꾸어야 한다. 즉, 다른 법칙으로 구조 기술 문자열을 생성하는 것이다. 그 법칙은 각 원자별로 결합구조를 표현하되, 각 레벨(Level, 경로의 길이, 깊

이)별로 그 레벨에 해당하는 원자들의 경로를 기술하여, 이를 정렬하여 저장하는 것이다. 즉, 그래프의 모든 에지에 가중치를 같게 주고, 각 원자별로 최소비용신장트리를 구성한 다음, 모든 원자까지의 경로를 깊이별로 나누어 기술하여 저장하는 것이다. 다시 말하면 한 중심원자에서 다른 원자까지의 최단경로를 기술하고, 이를 레벨별로 나눈 후, 각 레벨별로 경로를 기술한 문자열들끼리 알파벳 순서대로 정렬하여 정렬된 형태로 데이타베이스에 저장하는 것이다.

그림 5의 DB저장구조를 본 기법으로 저장하면 표 2와 같다.

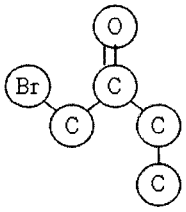


그림 5 DB 저장 구조

표 2에서, Br원자 레코드의 레벨 3인 ", c . c . c , c . c . e"를 설명하면, Br원자에서 최단경로의 길이가 3인 원자는 O원자와 C원자가 있으며, 각각 "c . c . e"와 "c . c . c"로 기술될 수 있다. 이를 알파벳순서로 정렬하고, 시작 위치를 나타내는 ','를 앞에 붙여서 구분하면 ", c . c . c , c . c . e"가 되는 것이다.

표 2 해결 방안의 DB 저장 형태

원자	레벨 1	레벨 2	레벨 3	레벨 4
Br	, c	, c . c	, c . c . c , c . c . e	, c . c. c . c
C	, b , c	, c . c , c . e	, c . c . c	
C	, c , c , e	, c . b , c . c		
O	, d	, d . c , d . c	, d . c . b , d . c . c	
C	, c , c	, c . c , c . e	, c . c . b	
C	, c	, c . c	, c . c . c , c . c . e	, c . c . c . b

마찬가지 방법으로 하여, 그림 6의 쿼리구조를 본 기법대로 기술하면 표 3과 같다.

마찬가지로 Br 원자 레코드의 레벨 3인 "% , c . c . e %"를 설명하면, 시작점을 알리는 ',' 앞에 와일드카드(%)를 붙이고, 경로를 기술한 "c . c . e"를 붙인 후, 마지막에 와일드카드(%)를 덧붙이는 것이다. 즉, ", c . c

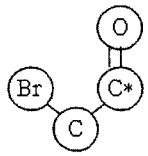


그림 6 쿼리 구조

표 3 해결 방안의 쿼리구조 기술 형태

원자	레벨 1	레벨 2	레벨 3	레벨 4
Br	% , c %	% , c . c %	% , c . c . e %	
C	% , b % , c %	% , c . e %		
C	% , c % , e %	% , c . b %		
O	% , d %	% , d . c %	% , d . c . b %	

. e"가 존재하는 레코드를 검색하라는 것이다. SQL문으로 표현하면 다음과 같다.

WHERE level1 LIKE '%,c%' AND level2 LIKE '%,c.c%' AND level3 LIKE '%,c.c.e%'

이렇게 기술하고 부분문자열검색을 하면 표 2의 Br원자 레코드의 레벨 3을 찾아낼 수 있다.

본 기법을 사용하면, 연결부분(RS3 시스템에서의 '*')을 지정하지 않아도 되고, 따라서 모든 구조를 기술할 수 있기 때문에, 정도가 좋아진다. 또한 전체 경로를 기술하고 이 경로끼리 알파벳 순서로 정렬하기 때문에, 동일원자의 동일결합에 대한 우선순위가 명확하여 재현율이 향상되게 된다.

4.2 해결 방안의 알고리즘 순서도

본 기법의 문자열 생성 알고리즘의 순서도는 그림 7과 같다.

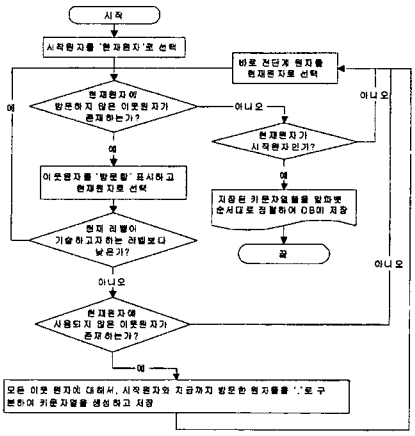


그림 7 해결 방안의 문자열 생성 알고리즘 순서도

5. 성능 평가

본 장에서는 Ullmann 알고리즘과 RS3 시스템, 그리고 개선된 RS3 시스템으로 부분구조검색을 수행하고, 그 결과를 가지고 각각의 성능을 평가한다.

5.1 실험 방법

672개의 약물 화학 구조들에 대해 그림 8의 15개의 부분구조(Moiety)를 가지고 검색을 수행하였다.

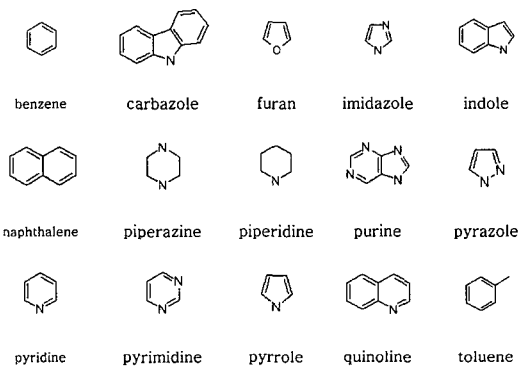


그림 8 질의할 부분구조

RS3 시스템을 실험할 경우, 모든 원자에 '*'가 붙은 것으로 하였다.

실험 환경은 펜티엄Ⅳ 1.6㎓, 메모리 1GB,하드디스크 80GB이며 Windows 2000 Server 운영체제에 데이타베이스는 MS SQL Server 2000을 사용하였다.

5.2 실험 결과

672개의 약물 화학 구조들에 대해 Ullmann 알고리즘과 RS3 시스템, 그리고 본 기법으로 수행시간, 정도, 재현율에 대해 실험하였다.

RS3 시스템과 본 기법의 실험에서는 DBMS의 검색 결과, 즉 핵심 엔진에서 나온 결과에서, 원자 종류별 개수를 질의의 원자 종류별 개수와 비교하여 적은 것은 제외시키는 과정을 거쳤다. 즉, 다차시간 알고리즘까지는 수행시켰다.

스크리닝 시스템(RS3, 본 기법)은 결과를 다시 ABAM (Atom-by-Atom Matching)을 하여 정확한 결과만을 출력하게 된다. 여기서는 ABAM에 Ullmann 알고리즘을 사용하여, 최종 결과가 나오기까지의 시간도 측정하였다.

수행시간의 결과는 표 4와 같다. "+ Ullmann"이 들어간 항목은, ABAM까지 수행한 결과를 나타낸다. 또한 평균은 전체 수행 시간을 부분구조의 개수로 나눈 값으로, 대략 1개의 부분구조를 검색하는데 소요되는 시간을 나타낸다.

표 4 수행 시간 [초]

부분구조	Ullmann	RS3 시스템		본 기법	
		RS3	+ Ullmann	본 기법	+ Ullmann
benzene	10	1	10	1	7
carbazole	11	1	12	3	3
furan	2	1	2	1	1
imidazole	1	1	1	1	1
indole	20	1	20	2	2
naphthalene	15	1	15	2	2
piperazine	11	1	12	1	11
piperidine	13	1	14	1	13
purine	6	2	7	1	1
pyrazole	1	1	1	1	1
pyridine	13	2	15	1	1
pyrimidine	6	2	7	1	1
pyrrole	7	1	8	1	1
quinoline	10	1	11	3	3
toluene	5	1	6	2	5
평균	8.7	1.2	9.4	1.5	3.5

RS3 시스템과 본 기법의 정도와 재현율의 결과는 표 5와 같다. 참고로 Ullmann 알고리즘은 정도와 재현율이 당연히 모두 100%이다.

표 5 정도와 재현율 [%]

부분구조	정도		재현율	
	RS3 시스템	본 기법	RS3 시스템	본 기법
benzene	72.5	98.5	100	100
carbazole	0.3	100	100	100
furan	1.4	46.7	100	100
imidazole	21.8	93.2	63.4	100
indole	2.5	84.6	100	100
naphthalene	0.6	25.0	100	100
piperazine	10.6	64.3	100	100
piperidine	11.9	48.8	100	100
purine	1.4	100	100	100
pyrazole	0.8	80.0	25.0	100
pyridine	8.2	82.2	100	100
pyrimidine	11.4	100	100	100
pyrrole	2.9	27.1	100	100
quinoline	0.2	100	100	100
toluene	61.2	98.9	100	100
평균	13.8	76.6	92.6	100

여기서 평균은 수행시간에서의 평균과 마찬가지로 대략 1개의 부분구조에 관한 정도 또는 재현율 값을 나타낸다.

5.3 결과 분석

본 기법으로 약물을 걸러낸 후 Ullmann 알고리즘으로 최종 결과를 출력하는 것이, 모든 약물에 Ullmann 알고리즘을 수행하는 것보다 2.5배 빨랐다. RS3는 정도

가 매우 낮기 때문에 수행 속도 향상을 얻지 못했다.

예상했던 바와 같이, RS3 시스템은 정도가 매우 낮았다. 이를 본 기법은 매우 많이 향상시켰다. 또한 RS3 시스템은 재현율에서 imidazole에서 63.4%, pyrazole에서 25.0%를 기록했다. 하지만 본 기법의 재현율은 언제나 100%이다.

5.4 본 기법의 재현율 검토

순수 그래프적인 측면에서 보면, 본 기법은 사이클의 일부분을 검색하려고 할 때에는 재현율이 100%가 되지 않는다. 하지만 화학구조에서 사이클은 사이클 자체로 의미가 있다. 즉, 사이클과 그 사이클의 일부분과는 화학적 특성이 확연히 달라지게 되므로, 사이클의 일부분을 검색하고자 하는 시도는 무의미하다. 이에, 본 기법은 화학 부분 구조 검색에서 100%의 재현율을 달성할 수 있을 것으로 보여진다.

5.5 저장 공간의 크기 비교

원자의 개수를 n이라 하고 두 시스템의 저장 공간을 비교하면 다음과 같다.

RS3 시스템은 하나의 원자당, 모든 원자를 표시하는 n개의 문자와 그 만큼의 '.'을 적어주어야 하므로, 총 저장 공간은 $n*(n+n)$해서 $2n^2$의 저장 공간을 필요로 한다.

본 기법은 원자들이 일렬로 늘어서 있을 때, 가장 긴 경로를 기술하여야 하므로 가장 많은 저장 공간을 필요로 한다. 이를 계산해보면, 하나의 원자당$(1+2+3+\cdots+n)$의 문자와 이에 상응하는 '.'을 기술해야 한다. 따라서 총 저장 공간은 $n*\{(1+2+3+\cdots+n)*2\}$해서 n^3+n^2이다. 이상을 정리하면 표 6과 같다.

실제로, 672개의 화학 구조를 저장한 RS3 시스템의 인덱스 크기는 3,840KB이고 본 기법은 10,266KB으로, 본 기법이 RS3 시스템보다 2.7배 많은 저장 공간을 요구한다. 좀 더 자세히 구조를 기술한 본 기법이 RS3 시스템보다 많은 저장 공간을 필요로 한다.

표 6 저장 공간의 비교 (n: 총원자수)

RS3	본 기법
$2n^2$	최대 n^3+n^2

5.6 본 기법의 활용 방안

현재 구축된 약물/타겟 데이터베이스(http://166.104.115.14)의 부분구조 검색서비스로 본 모듈을 사용할 계획이다.

이때 본 기법의 문제점인 저장 공간의 증가도 개선할 예정이다. 중복된 문자열들을 한 문자로 치환한다면 저장 공간을 많이 절약할 수 있을 것으로 보인다. 예를 들어 ", c . c . c , c . c . e"에서 중복되는 ", c . c . "을 '~'으로 치환하여 ", c . c . c , ~ e" 등으로 나타내는 것이다.

6. 결 론

본 논문에서는 약물의 화학 부분 구조를 검색하는 RS3 시스템의 문제점을 규명하고, 이의 개선 방안을 제시하였다.

RS3 시스템은 화학 구조를 정해진 규칙에 따라 문자열로 기술하여, 부분구조검색 문제를 부분문자열검색으로 해결한 시스템이다. 하지만 이때 쓰인 규칙에는 치명적인 단점이 있어, 재현율과 정도를 많이 떨어뜨린다.

이를 개선한 본 기법은 저장 공간이 증가하는 단점이 있으나, 재현율과 정도를 급격히 향상시킨다.

앞으로는 본 기법의 정도를 더욱 향상시키는 방안 및 저장 공간 축소 방안에 대하여 연구할 계획이다.

참 고 문 헌

[1] Alfred Burger, A Guide to the Chemical Basis of Drug Design, John Wiley & Sons Inc., July 1983.

[2] R. C. Read and D. G. Corneil, "The graph isomorphism disease," J. Graph Theory, 1, 339-363, 1977.

[3] P. G. Dittmar, N. A. Farmer, W. Fisanik, R. C. Haines, J. Mockus, "The CAS ONLINE Search system 1. General system design and selection, generation, and use of search screens," Journal of Chemical Information and Computer Sciences, vol.23, no.3, pp.93-102, 1983.

[4] Daylight, http://www.daylight.com, Daylight Chemical Information Systems, Inc., 27401 Los Altos, Suite 370, Mission Viejo, CA 92691, USA.

[5] G. A. Hopkinson, "The Accord Component Software Approach," J. Chem. Inf. Comput. Sci., 37, 143-145, 1997.

[6] L. C. Ray and R. A. Kirsch, "Finding chemical records by digital computers," Science, 126, 814-819, 1957.

[7] J. R. Ullmann, "An algorithm for subgraph isomorphism," Journal of ACM, vol.23, 31-42, 1976.

[8] M. F. Lynch, "R&D in chemical information science : Retrospect and prospect," Chemical Structures : The international language of chemistry, W. A. Warr ed., pp.1-10, Springer-Verlag, 1988.

[9] W. Graf, H. K. Kaindl, H. Kniess, and R. Warszawski, "The third BASIC fragment search dictionary," J. Chem. Inf. Comput. Sci., 22, 177-181, 1982.

[10] A. P. Johnson and A. P. Cook, "Automatic keyword generation for reaction searching," 'Modern Approaches to Chemical Reaction Searching,' ed. P. Willett, Gower, Aldershot, pp. 184-193, 1985.

[11] R. J. Feldmann, G. W. A. Milne, S. R. Heller, A.

Fein, J. A. Miller, and B. Koch, "An interactive substructure search system," J. Chem. Inf. Comput. Sci., 17, 157-163, 1977.
[12] R. Attias, "DARC substructure search system : a new approach to chemical information," J. Chem. Inf. Comput. Sci., 23, 102-108, 1983.
[13] Z. M. Nagy, S. Kozics, T. Veszpremi, and P. Bruck, "Substructure Search on Very Large Files Using Tree-structured Databases," 'Chemical Structures: The International Language of Chemistry,' ed. W. A. Warr, Springer-Verlag, Heidelberg, pp. 127-130, 1988.
[14] Z. M. Nagy, "How can parallel algorithms help to find new sequential algorithms?," J. Chem. Inf. Comput. Sci., 33, 542-544, 1993.
[15] A. Bartmann, H. Maier, D. Walkowiak, B. Roth, and M. G. Hicks, "substructure searching on very large files by using multiple storage techniques," J. Chem. Inf. Comput. Sci., 33, 539-541, 1993.
[16] RS3, http://www.accelrys.com/rs3
[17] J. Moore and J. R. Hoover, US Patent 5 577 239, 1996.

이 환 구

2002년 2월 한양대학교 기계공학부 졸업 2004년 2월 한양대학교 정보통신대학원 소프트웨어전공 졸업. 2004년 2월~현재 한국전자통신연구원 임베디드S/W연구단 무선인터넷플랫폼연구팀 근무

차 재 혁

1987년 서울대학교 계산통계학과 졸업 (이학사). 1991년~1997년 서울대학교 컴퓨터공학과 석사와 박사학위 취득. 1997년~1998년 한국학술진흥재단부설 첨단학술정보센터 선임연구원. 1998년~2001년 한양대학교 사범대학 컴퓨터교육과 전임강사. 2001년~현재 한양대학교 정보통신대학 정보통신학부 조교수. 관심분야는 XML 데이타베이스, 플래시메모리 기반 저장시스템, e-learning, 멀티미디어 콘텐츠 적응화임