완전이진트리의 이항트리에 대한 임베딩

윤수민*, 최 정**, 임형석*
*전남대학교 전산학과
**기전여자대학 사무자동학과

Embedding Complete binary trees in Binomial trees

Su-Min Youn*, Jung Choi**, Hyeong-Seok Lim*
*Dept. of Computer Science, Chonnam National Univ.
**Dept. of Office Automation, Kijeon College

Abstract – Trees are the underlying structure for divide-and-conquer algorithms and the graphs that provide the solution spaces for NP-complete problems. Complete binary trees are the basic structure among trees. Therefore, if complete binary trees can be embedded in binomial trees, the algorithms which are provided by complete binary trees can be performed efficiently on the interconnection networks which have binomial trees as their subgraphs or in which binomial trees can be embedded easily.

In this paper, we present dilation 2 embedding of complete binary trees in binomial trees.

1. 서론

최근의 컴퓨터 시스템은 하드웨어 기술의 발달과 함께 단일 프로세서 중심의 처리구조에서 점차 여러 개의 프로세서를 상호 연결하여 처리하는 병렬처리 컴퓨터 구조로 발전해 가고 있다. 이러한 추세에 따라 다양한 형태의 병렬처리 컴퓨터 구조들이 개발되고 이들 구조에서 여러 가지 문제들을 풀기 위한 많은 병렬 알고리즘들이 설계되었다. 이러한 알고리즘들을 원래와는 다른 연결망 구조에서 실행시킬 수 있는지는 병렬처리에서 중요한 문제중 하나이다. 이러한 방법 중에서 널리 쓰이는 것으로 임베딩이 있다[1, 2, 6, 7].

그래프 G의 그래프 H에 대한 임베딩 $f=(\phi,\rho)$는 다음과 같이 정의한다. ϕ는 G의 정점 v에서 H의 정점 $\phi(v)$로의 함수이고, ρ는 G의 에지 $e = (v, w)$에서 $\phi(v)$와 $\phi(w)$를 잇는 H의 경로 $\rho(e)$로의 함수이다. 임베딩의 비용을 측정하는 척도로는 연장률(dilation), 밀집률(congestion), 확장률(expansion) 등이 있다. G의 에지 e의 연장률은 H상에서의 경로 $\rho(e)$의 길이를 말하고, 임베딩 f의 연장률은 G의 모든 에지의 연장률 중 최대값이다. H의 에지 e'의 밀집률은 e'를 포함하는 $\rho(e)$의 개수를 말한다. 임베딩 f의 밀집률은 H의 모든 에지의 밀집률의 최대값이다. 임베딩 f의 확장률은 G의 정점의 개수에 대한 H의 정점의 개수의 비를 말한다.

이러한 임베딩 문제중 트리 구조를 임베딩하는 문제는 특별한 중요성을 띄고 있다. 트리는 자료구조 또는 분할정복(divide-and-conquer) 알고리즘의 기반이 되거나 NP-complete 문제의 solution space를 제공하는 구조로서 널리 이용되고 있기 때문이다.

트리 중에서 가장 기본적인 구조인 완전 이진 트리에 대한 임베딩 결과로는 하이퍼큐브에 연장률 2, 확장률 1 또는 연장률 1, 확장률 2로 임베딩할 수 있음이 알려져 있으며[3], 완전 이진 트리가 재귀원형군 $G(N, 4)$의 부그래프임이 알려져 있다[4]. 방송 알고리즘의 기본이 되는 구조인 이항 트리는 재귀원형군

$G(N, 4)$의 부그래프이고 하이퍼큐브의 부그래프임이 이미 알려져 있다[3, 5]. 그렇지만 완전 이진 트리를 이항 트리에 임베딩하는 효율적인 방법은 개발되어 있지 않다. 본 논문에서는 완전 이진 트리를 이항 트리에 임베딩하는 효율적인 방법을 재시한다.

본 논문의 구성은 다음과 같다. 2장에서는 정의 및 기본 성질에 대해 설명하고 3장에서는 완전 이진 트리를 이항 트리에 임베딩하는 방법을 제시하고 4장에서는 결론을 맺고 추후 연구 방향을 살펴본다.

2. 정의 및 기본 성질

완전 이진 트리와 이항 트리의 정의는 다음과 같다.

정의 1 높이가 k인 **완전 이진 트리** BT_k
공집합이거나 루트와 왼쪽 서브트리, 오른쪽 서브트리라고 부르는 두 개의 분리된 이진 트리로 구성된 정점의 유한집합이다.

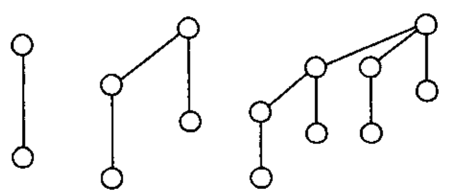
높이가 k인 완전 이진 트리는 $2^{k+1}-1$개의 정점을 갖는다.

정의 2 높이가 k인 **이항 트리** B_k

(1) B_0는 하나의 정점 r_0로 구성되며 r_0가 B_0의 루트이다.

(2) $k \geq 1$에 대해, $B_{(k-1)l}$과 $B_{(k-1)r}$은 서로 소인 B_{k-1}이고 $B_{(k-1)l}$의 루트를 $B_{(k-1)r}$의 가장 왼쪽 자식 정점이 되도록 에지를 추가함으로써 얻어지는 트리이다.

이러한 이항 트리에 대해 다음 성질이 성립함을 쉽게 알 수 있다.



(a) B_1 (b) B_2 (c) B_3

그림 1. 이항 트리의 예

성질 1 $B_k(k \geq 0)$의 정점의 개수는 2^k이다.

성질 2 $B_k(k \geq 0)$의 각 정점에 한 개의 자식 노드를 추가한 트리는 B_{k+1}이다.

성질 3 $B_k(k \geq 1)$에서 r^k를 제거하면 $B_0, B_1, \cdots, B_{k-1}$이 각각 하나씩 존재하는 포리스트가 된다.

3. 완전이진트리의 이항트리에 대한 임베딩

이항 트리를 이진 트리 형태로 변형한 트리는 완전 이진 트리를 부그래프로 가지고 있다. 이진 트리 형태로 변환된 이항 트리가 연장률 2로 이항 트리에 임베딩 가능하다면 완전 이진 트리가 이항 트리에 연장률 2로 임베딩 가능하다.

이항 트리를 이진 트리로 변형하는 방법은 다음과 같다. 임의의 노드(c)의 자식 노드들을 $c_1, \cdots, c_n$이라 하자. c에서 가장 왼쪽 자식(c_1)을 제외한 나머지 자식 $c_i(i=2,\cdots,n)$는 부모 노드(c)와의 에지를 제거하고 c_{i-1}과 에지를 갖는다.

완전 이진 트리 BT_k에 정점 하나가 추가된 트리는 다음과 같이 정의한다.

정의 T_k는 완전 이진 트리 BT_k의 루트 r_k와 하나의 정점 r_k'가 인접한 형태의 트리이고 r_k가 T_k의 루트이다.

정리 1 이항 트리 B_{k+1}를 이진 트리로 변형한 트리는 T_k이다.

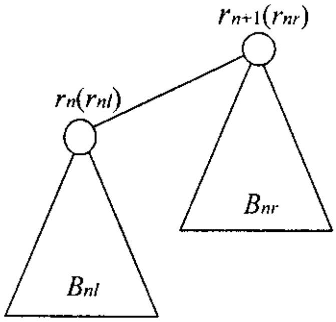
증명 증명은 수학적 귀납법을 사용한다. $k=1$인 경우 이항 트리 B_2를 이진 트리로 변형한 트리는 T_1이 됨을 쉽게 알 수 있다. $k < n$일 때 위 정리가 성립한다고 가정하고, $k=n$일 때 위 정리가 성립됨을 보이고자 한다.

B_{n+1}은 두 개의 동형인 $B_{nl}(B_{nr})$에 대해 왼쪽 서브트리 B_{nl}의 루트 r_{nl}과 오른쪽 서브트리 B_{nr}의 루트 r_{nr}사이에 하나의 에지가 존재한다(그림 2(a)).

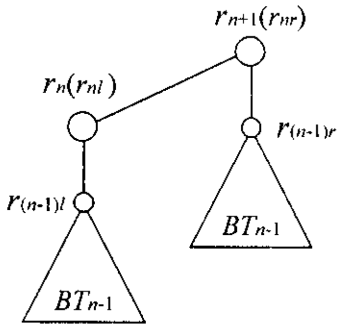
가설에 의해 각 $B_{nl}(B_{nr})$을 이진 트리로 변형한 트리는 $T_{(n-1)l}(T_{(n-1)r})$이 되고 이것은 그림 2의 (b)와 같이 BT_{n-1}의 루트 $r_{(n-1)l}(r_{(n-1)r})$과 하나의 노드가 인접한 형태가 된다.

B_{n+1}중에서 r_{nl}, r_{nr}, $r_{(n-1)l}$, $r_{(n-1)r}$ 부분만 이진 트리로 변형하면 된다. $r_{(n-1)r}$은 r_{nr}과 에지를 제거하고 r_{nl}과 에지를 갖는다. r_{nl}에 대해 왼쪽 자식 $r_{(n-1)l}$, 오른쪽 자식

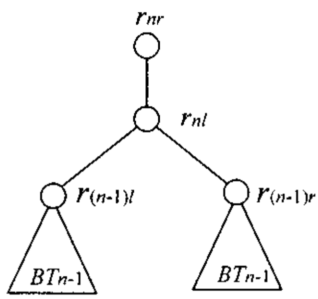
$r_{(n-1)r}$은 완전 이진 트리 BT_n이 되며 r_{nl}과 인접한 r_{nr}과 함께 T_n이 된다(그림 2(c)). □



(a) B_{n+1}



(b) B_{n+1}



(c) T_n

그림 2. 이항 트리 B_{n+1}과 T_n

정리 2 다음 조건 (a)와 (b)를 만족하는 $T_k(k \geq 1)$의 이항 트리 B_{k+1}에 대한 연장률 2인 임베딩이 존재한다.

(a) T_k의 부트리 BT_k에 속하는 어떤 정점도 B_{k+1}의 루트에 사상되지 않는다.

(b) T_k의 루트는 B_{k+1}의 루트와 연결된 B_k의 루트에 사상된다.

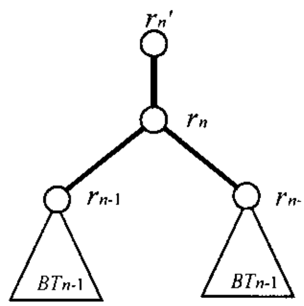
증명 증명은 수학적 귀납법을 사용한다. k=1인 경우 T_1의 루트 r_1을 이항 트리 B_2의 루트와 연결된 r_1에 사상시키고 T_1의 $r_1{}'$를 B_2의 루트 r_2에 사상시키면 조건 (a)와 (b)를 만족한다.

$k<n$인 경우 위 정리가 성립한다고 가정하고, $k=n$일 때 위의 두 조건을 만족하면서 T_n이 B_{n+1}에 연장률 2로 임베딩 됨을 보이고자 한다.

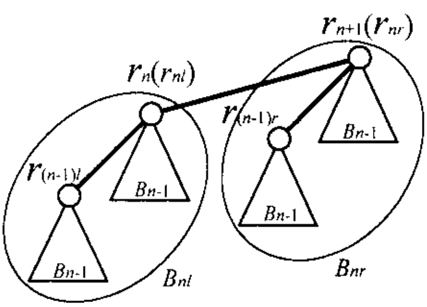
B_{n+1}은 두 개의 동형인 B_{nl}과 B_{nr}에 대해 왼쪽 서브트리 B_{nl}의 루트 r_{nl}과 오른쪽 서브트리 B_{nr}의 루트 r_{nr} 사이에 하나의 에지가 존재한다(그림 3(b)). 가설에 의해 T_{n-1}은 $B_{nl}(B_{nr})$에 연장률 2로 임베딩 가능하고 T_{n-1}의 루트는 조건 (b)에 의해 $r_{(n-1)l}$과 $r_{(n-1)r}$에 사상되고 조건 (a)에 의해 r_{nl}과 r_{nr}에는 T_{n-1}의 부그래프인 BT_{n-1}의 어떤 정점도 사상되지 않는다.

T_n의 루트는 이항트리 B_{n+1}의 루트와 연결된 r_{nl}에 사상되고 $r_n{}'$는 B_{n+1}의 루트 r_{nr}에 사상하여 조건 (a)와 (b)를 만족한다.

B_{n+1}에서 r_{nl}과 $r_{(n-1)l}$ 사이와 r_{nl}과 r_{nr} 사이에는 에지가 존재하며 r_{nl}와 $r_{(n-1)r}$ 사이에는 길이가 2인 경로가 존재한다. 그러므로 조건 (a)와 조건 (b)를 만족하는 T_n의 B_{n+1}에 대한 연장률 2 인 임베딩이 존재한다(그림 3). □



(a) T_n



(b) B_{n+1}

그림 3. T_n의 B_{n+1}에 대한 임베딩

정리 1과 정리 2에 의해 완전 이진 트리는 이항 트리에 연장률 2로 임베딩 가능하다.

정리 3 완전 이진 트리 $BT_k(k \geq 1)$는 이항 트리 B_{k+1}에 연장률 2로 임베딩 가능하다.

증명 완전 이진 트리 BT_k는 T_k의 부트리이고 정리 2에 의하면 T_k는 B_{k+1}에 연장률 2로 임베딩 가능하다. 그러므로 완전 이진 트리 BT_k는 이항 트리 B_{k+1}에 연장률 2로 임베딩 가능하다. □

4. 결론

본 논문에서는 완전 이진 트리를 이항 트리에 임베딩하는 방법을 제시하였다. 이항 트리 B_{k+1}을 이진 트리로 표현하면 완전 이진 트리의 루트와 하나의 노드가 인접한 T_k가 됨을 보였고, T_k는 이항 트리 B_{k+1}에 연장률 2로 임베딩 가능함을 보였다. 완전 이진 트리 BT_k는 T_k의 부그래프가 되는 결과를 적용하면 BT_k를 B_{k+1}에 연장률 2로 임베딩할 수 있다.

따라서 특정 연결망에서 두 트리가 제공하는 효율적 알고리즘을 수행하고자 할 때 이항 트리를 임베딩할 수 있다면 연장률 2의 관계로 완전 이진 트리가 제공하는 알고리즘을 수행할 수 있다.

앞으로 완전 이진 트리를 이항 트리에 연장률 1로 임베딩하는 방법과 이항 트리를 완전 이진 트리에 효율적으로 임베딩하는 방법에 관한 연구가 필요하리라 생각된다.

참고문헌

[1] S. B. Akers, and B. Krishnamurthy, "A group-theoretic model for symmetric interconnection networks," IEEE Trans. Comput., Vol. 38, pp. 555-566, 1989.

[2] N. Bagherzadeh, M. Dowd, and N. Nassit, "Embedding an arbitrary binary tree into the star graph," IEEE trans. Comput. Vol.45,, pp. 475-481, 1996.

[3] S. N. Bhatt, F. R. K. Chung, F. T. Leighton, and A. L. Rosenberg, "Efficient embedding of trees in hypercubes," SIAM J. Comput. Vol. 21, pp. 151-162, 1991.

[4] H.-S. Lim, J.-H. Park, and K.-Y. Chwa, "Embedding Trees in Recursive Circulants," Discrete Applied Mathematics 69, pp. 83-99, 1996.

[5] J. Q. Michael, Parallel Computing theory and practice, pp. 137-139, 1994.

[6] A. Wagner, "Embedding arbitrary binary trees in a hypercube," J. Parallel and Distrib. Comput. 7, pp. 503-520, 1989.

[7] A. Y. Wu, "Embedding of tree networks into hypercubes," J. Parallel and Distrib. Comput., Vol. 2, pp. 238-249, 1985.