

# 어떤 정규언어와 Prefix Coding

Mordecai Golin\*, 정회원 나 현 숙\*\*

## Some Regular Languages and Prefix Coding

Mordecai Golin\*, Hyeon-Suk Na\*\* *Regular Member*

### 요 약

코드는 단어들의 집합이다 어떤 단어도 다른 것의 Prefix가 아닌 단어들의 집합을 Prefix(-Free) Code라 부르고, 여기서 Prefix Coding이란, 주어진 수  $n$ 에 대하여,  $n$ 개의 단어로 이루어진 Prefix 코드들 중에서 단어길이의 총합이 최소인 최적 Prefix 코드를 찾는 것을 말한다. 이 논문에서는 이의 특수한 경우로서 특정 정규언어군에 속하는 임의의 정규언어  $L$ 에 대하여,  $L$ 에 속하는 Prefix 코드들 중 최적의 코드를 찾는 방법을 제시한다 또, 수  $n$ 이 변함에 따라 최적 Prefix 코드의 구조가 어떻게 변하는지 그 성질을 트리구조를 이용해 밝힌다

**Keywords :** Coding, Prefix Code, Regular Language, Finite Automata

### ABSTRACT

Code is a set of words. If, for any pair of words in the code, one is not prefix of another, the code is called "Prefix(-Free) Code". The prefix coding problem is, given  $n$ , to find an optimal code with the minimum-sum of lengths of  $n$  words. As a special case of this, we present a method to find, given language  $L$  in some specific classes of regular languages, an optimal code among prefix codes in  $L$ . We also show how the structure of optimal codes varies as  $n$  grows, using trees.

### I. 서 론

우선 몇 가지 용어를 소개하면서 시작하기로 하자.  $\Sigma$ 는  $\{0,1\}$ 이나  $\{a,b,c\}$ 와 같은 2개 이상의 기호로 이루어진 알파벳이다 코드  $C = \{w_1 \dots w_n\}$ 는 이 알파벳의 기호들로 이루어진 단어들의 집합이다. 만일 한 단어  $w = \sigma_{i_1} \sigma_{i_2} \dots \sigma_{i_n}$ 가 다른 단어  $w' = \sigma_{j_1} \sigma_{j_2} \dots \sigma_{j_m}$ 의 시작이 되면, 즉,  $m \geq n$ 이면서 모든  $k \leq n$ 에 대해서  $\sigma_{i_k} = \sigma_{j_k}$ 이면,  $w$ 는  $w'$ 의 prefix라고 한다. 예를 들어, 01은 010011의 prefix이다. 주어진 코드  $C$ 에 대해서, 그것의 구성원인 어떤 단어  $w$ 도 다른 단어  $w'$ 의 prefix가 아닐 때, 우리

는 이 코드  $C$ 가 Prefix-Free하다고 하고, Prefix 코드라고 부른다. 주어진 이산 확률 분포  $P_n = \{p_1, \dots, p_n\}$  ( $0 \leq p_i \leq 1$ ,  $\sum_{i=1}^n p_i = 1$ )와  $n$ 개의 원소를 갖는 코드  $C_n$ 에 대해서,  $cost(C_n, P_n) = \sum_{i=1}^n p_i |w_i|$ 로 정의된다. 즉,  $cost(C_n, P_n)$ 는 코드  $C_n$ 의 단어들의 확률분포  $P_n$ 에 따른 평균 길이로 정의된다. 주어진 확률분포  $P_n$ 에 대해서,  $cost(C_n, P_n)$ 을 최소화하는 Prefix 코드를 찾는 것은 흔히 Prefix coding 문제라고 불린다<sup>[1]</sup>. 특히, 확률분포  $P_n$ 이 균일할 때, 즉  $\forall i, p_i = 1/n$ 일 때에는  $\sum_{i=1}^n |w_i|$ 가 최소인 코드가 최적의 코드가 되며

\* Dept. Computer Science, HKUST (golin@cs.ust.hk), \*\* 숭실대학교 정보과학대학 컴퓨터학부 (hsnaa@computing.ssu.ac.kr)

논문번호: KICS 2004-08-166, 접수일자: 2004년 8월 26일

※ 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음

이 논문에서 우리는 이 경우만을 고려할 것이다 이 제 확률분포  $P_n$ 이 균일한 분포로 고정되었으므로, 앞으로  $cost(C_n, P_n)$ 을 단순히  $cost(C_n)$ 로 표기하기로 한다. 이 논문에서 우리가 다루는 문제는 다음과 같다.

문제1: 주어진  $n$ 과  $\Gamma \subseteq \Sigma^*$ 에 대하여,  $C_n \subseteq \Gamma$ 이면서  $cost(C_n)$ 을 최소화하는 prefix 코드  $C_n$ 을 찾아라.

예를 들어,  $\Gamma = (0+1)^*1$  일 때, 모든 prefix 코드  $C_n \subseteq \Gamma$  중에서  $cost(C_n)$ 을 최소화하는 코드  $C_n$ 을 찾는 것이 이 논문에서 다루는 문제이다. 특히, 이 문제는 자기 동기화 (self-synchronizing) 코드의 디자인에 응용될 수 있기 때문에 이미 많은 연구자들의 관심을 받아왔다<sup>3,4)</sup>. 우리는 이 문제를 임의의 패턴  $\Gamma$ 로 끝나는 단어들에 대해서 최적의 prefix 코드를 찾는 문제, 즉,

$C_n = \{w_i : w_i \in \Sigma^*\Gamma\}$  일 때,  $cost(C_n)$ 을 최소화하는 Prefix 코드를 찾는 문제로 일반화시켜 풀 뿐만 아니라, 더 넓은 범위의 언어특정한 종류의 DFA(Deterministic Finite Automata)에 의해서 받아들여지는 정규언어들(Regular languages), II.2절-에 대해서도 풀 것이다. 우리의 조건을 만족하는 정규언어  $\Gamma$ 에 대해서 최적 Prefix 코드  $C_n$ 은  $\Gamma$ 를 받아들이는 DFA  $M_\Gamma$ 에 관련된 특정한 상수뒤에서 정의될  $h_{M_\Gamma}$ -에만 의존함이 보여질 것이다. 따라서, 정규언어  $\Gamma$ 와 그것의 DFA  $M_\Gamma$ 가 주어지면, 이 오토마타에 의존하는 상수  $h_{M_\Gamma}$ 을 계산할 수 있고, 이 상수를 이용해 최적의 prefix 코드  $C_n$ 의 구조를 알 수 있는 것이다.

앞으로 II장에서는 우리가 관심있는 정규언어군과 그것의 DFA, 각 DFA에 대응되는 무한트리에 대해 소개하고, Prefix coding 문제를 최소 외부경로길이 (minimum external path length)를 갖는 트리를 찾는 문제로 전환시킨다. III장에서는 필요한 정의와 도구들에 대해서 소개한 뒤, 우리의 주요 결과와 그 증명을 제시한다.

## II. DFA, 정규언어, 그리고 무한 트리

### 2.1 DFA와 정규언어

DFA  $M$ 은 5개의 원소를 갖는 순서쌍  $(Q, q_0, F, \Sigma, \delta)$ 로 정의되는데[5], 여기서  $Q$ 는 State들의 유한집합,  $q_0 \in Q$ 는 Start state,  $F \subseteq Q$ 는 Accepting state들의 집합,  $\Sigma$ 는 알파벳, 그리고  $\delta : Q \times \Sigma \rightarrow Q$ 는 Transition 함수이다.  $M$ 은 하나의 Start state  $q_0$ 와 Accepting state들을 포함한 State들을 노드로 가지면서 각 노드로부터 정확히  $|\Sigma|$ 개의 간선들 각 간선이 하나의 알파벳 기호에 대응-이 나가는 Directed graph로 표현되기도 한다. 하나의 단어  $w \in \Sigma^*$ 가 입력되면, Start state  $q_0$ 로부터 시작하여 한번에 하나의 알파벳씩  $w$ 로부터 읽어 들이면서 이에 상응하는 간선을 따라 새로운 State로 옮겨간다.  $w$ 의 마지막 알파벳을 읽어 들인 후  $M$ 이 Accepting state에 있게 되면, 단어  $w$ 는 DFA  $M$ 에 의해 받아들여진다고 말한다. DFA  $M$ 에 의해 받아들여지는 단어  $w$ 의 집합을  $L(M)$ 이라고 하고, 이렇듯 적당한 DFA에 의해 받아들여지는 언어들을 정규언어라고 부른다.

### 2.2 어떤 정규언어군과 그것의 DFA

그러면 이제 본격적으로 우리가 관심있는 언어군과 그것의 DFA에 대해 살펴보기로 하자.

- L1군:** 임의의 패턴  $\Gamma \in \Sigma^*$ 에 대해서, 이 패턴으로 끝나는 단어들의 집합, 즉  $L = \Sigma^*\Gamma$
- L2군:** 임의의 패턴  $\Gamma \in \Sigma^*$ 에 대해서, 이 패턴을 포함하는 단어들의 집합, 즉  $L = \Sigma^*\Gamma\Sigma^*$
- L3군:** 주어진  $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_k\}$ 와  $(r_1, r_2, \dots, r_k)$ ,  $r_i$ 는 음이 아닌 정수들, 에 대하여  $L = (\sigma_1^{r_1} + \sigma_2^{r_2} + \dots + \sigma_k^{r_k})^*$

그림1(a)와 그림2(a)는 L1군에 포함되는 언어들에 대응하는 DFA로서, '01'로 끝나는 단어를 받아들이는 DFA  $M_1$ 과 'abca'로 끝나는 단어를 받아들이는 DFA  $M_2$ 이다. L2군에 포함되는 언어의 DFA는 모두 L1군의 DFA로부터 얻어질 수 있으므로 (L1군의 DFA의 Accepting state에서 나가는 모든 간선을 자신을 향하도록 함) 별도의 예는 생략한다. 그림 3(a)은 L3군에 속하는 언어  $(0^2 + 1^3)^*$ 의 DFA이다. L3군은 흔히 Varn Language라고도 불리는데, 알파벳 기호들  $\sigma_1, \sigma_2, \dots$ 들이 서로 다른 길이 혹은 비용

을 요구할 때, 이 알파벳을 이용한 Prefix Coding 문제, 즉 Varn Coding 문제<sup>6,7,8,9,10</sup>는 바로 L3군 혹은 Varn Language을 이용한 문제로 해석될 수 있음에 주목할 필요가 있다. 이상의 예들에서 알 수 있듯이, L1군, L2군, L3군들 모두 “하나의 Accepting state”를 갖는 DFA에 의해 받아들여지는 정규언어들이다.

### 2.3 트리문제로의 변환

알파벳  $\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_k\}$ , DFA  $M$ , 그리고 정규언어  $L = L(M)$ 을 가정하자. 또한, 다음과 같은 무한  $k$ -ary 트리를  $T_M$ 이라 하자; 루트는 빈 문자열(empty string)에 대응되고, 모든 노드는 정확히  $|\Sigma|$ 개의 밖으로 나가는 간선들 각각 알파벳의 기호  $\sigma_i \in \Sigma$ 에 대응을 갖고, 각 노드는 루트로부터 이 노드에 도달하기까지 거친 경로(path)에 의해 형성된 단어에 대응된다. 그러면,  $T_M$ 의 노드와  $\Sigma^*$ 의 단어사이엔 일대일 대응관계가 성립하며 각 노드는 그것에 대응하는 단어를 읽은 뒤 DFA  $M$ 이 위치한 State를 라벨로 갖는다.  $\Sigma^*$ 의 임의의 단어  $w$ 에 대응하는  $T_M$ 의 노드를  $u$ 라 하자.  $w$ 가 언어  $L$ 에 포함되는 것은 노드  $u$ 의 State가 Accepting state인 것과 동치이다. 그림1(b), 그림2(b), 그림3(b)는 모두  $T_M$ 의 노드들에 그것의 라벨 즉 그것의 State를 표시해 놓은 것이다. 주목해야 할 것은, II.2절에서 정의한 정규언어군들은 모두 하나의 Accepting state를 갖는 DFA를 가지므로, 주어진  $T_M$ 에서 Accepting state를 루트로 해서 뻗어 나오는 부분트리들은 모두 같은 구조를 갖는다는 것이다.  $depth(u)$ 는 루트로부터  $u$ 까지의 경로의 길이를 말하며, 따라서  $depth(u) = |w|$ 를 만족한다. 마지막으로, 우리는 정의1에서  $T_M$ 의 노드를 DFA  $M$ 에 의해 받아들여지는지 여부에 따라 “참노드”와 “거짓

정의1:  $M$ 은 주어진 DFA이고,  $T_M$ 는  $M$ 의 무한  $k$ -ary 트리이다.  $T_M$ 상의 임의의 노드  $u$ 에 대하여,  $u$ 의 State가 Accepting state이면 “참노드”, 그렇지 않으면 “거짓노드”라 부른다. 주어진 유한트리  $T$ 에서 참노드가 Leaf이면 “참leaf”, 내부노드이면 “참내부노드”라고 부르고, “외부경로길이” 혹은 “ $cost(T)$ ”란  $T$ 의 참leaf들의 Depth의 합을 말한다.

노드”로 나누고,  $T_M$ 의 임의의 부분트리  $T'$ 에 대해서 그것의 외부경로길이(external path length) 또는  $cost(T')$ 을 정의한다.

단어  $w, w'$ 에 대응하는 노드를 각각  $u, u'$ 이라 할 때, 단어  $w'$ 이 단어  $w$ 의 Prefix인 것은, 노드  $u'$ 이 루트로부터 노드  $u$ 에 이르는 경로위에 있는 것과 동치이다. 그러므로, 어떤 트리  $T' \subseteq T_M$ 의 참leaf들의 집합은  $C \subseteq L(M)$ 인 Prefix 코드  $C$ 에 대응된다. 또,  $cost(T') = cost(C)$ 를 만족한다. 역으로, 주어진 Prefix 코드

$C_n = \{w_1, w_2, \dots, w_n\} \subseteq L(M)$ 에 대해서, 이들을 참leaf로 갖는 트리  $T' \subseteq T_M$ 는 적어도 하나 존재하며,  $cost(C_n) = cost(T')$ 을 만족한다. 따라서, 앞장의 문제1-Prefix coding 문제는 아래의 문제2-최적트리문제로 변환된다.

문제2: 주어진  $n$ 과 DFA  $M$ 에 대해서,  $n$ 개의 참leaf를 갖는 트리  $T' \subseteq T_M$  중에서 최소의 외부 경로길이 혹은  $cost(T')$ 을 갖는 트리  $T' := T_n$ 을 찾아라.

그림4는 그림1의 DFA  $M_1$ 와  $T_{M_1}$ 에 대해, 각각 5개의 참leaf를 갖는 두 부분트리  $T', T'' \subseteq T_{M_1}$ 의 Cost들과 대응하는 Prefix 코드들을 비교한 것이다. 그림5는 DFA  $M_1$ 에 대해 외부경로길이 혹은 Cost를 최소화하는 최적트리들을  $T_{12}$ 부터  $T_{40}$ 까지를 나열해 본 것이다. 이 최적트리열을 유심히 살펴보면 (특히  $T_{21}$ 에서  $T_{22}$ 로,  $T_{23}$ 에서  $T_{24}$ 로 옮겨가는 과정), 이미 알고 있는  $T_{n-1}$ 로부터  $T_n$ 을 구하기 위해서는, 탐욕적인 방법 아직 쓰이지 않은 참노드 중에서 가장 Depth가 작은 노드를 선택하여 leaf로 보태는 방법 외에도, 참leaf 중에서 Depth가 가장 큰 것들을 제거하고, Depth가 가장 작은 참leaf를 내부노드(internal node)로 만들고 그것의 참자식들을 leaf로서 보태는 방법도 있다는 것과 언제 탐욕적인 방법을 써야하고 언제 비탐욕적인 방법을 써

정의2:  $u$ 와  $v$ 는  $T_M$ 의 참노드라 하자.  $v$ 가  $u$ 의 조상이면서  $u$ 와  $v$ 사이에 다른 참노드가 없으면  $v$ 는  $u$ 의 참부모,  $u$ 는  $v$ 의 참자식이라 하자.  $u$ 의 조상이면서 참노드가 없는 경우에는 루트를  $u$ 의 참부모로 하자.

야하는지 무언가 규칙이 숨겨져 있다는 것을 알 수 있다. 이 숨은 규칙을 밝히는 게 이 논문의 주요결과이며, 다음 장에서 소개한다

그림4(a)의 트리  $T'$ 을 보자.  $T'$ 은 모두 7개의 참노드를 갖고 있는데, 그 중  $S_1$ 과  $S_3$ 는 참내부노드,  $S_2, S_4, S_5, S_6, S_7$ 는 참leaf이다. 노드  $S_1$ 과  $S_3, S_5$ 와  $S_7$ 사이에, 참부모-참자식 관계가 성립한다 그러나,  $S_1$ 과  $S_7$ 사이에는 참부모참자식 관계가 성립하지 않는데, 이들 사이에 참노드  $S_3$ 이 있기 때문이다

### III. 주요한 결과

이제부터  $L$ 은 II.2절에서 정의한 정규언어군  $L_1, L_2, L_3$ 군에 속하는 임의의 정규언어를 가르키고,  $M$ 은 그것을 받아들이는 DFA를 가르킨다.

#### 3.1 몇가지 정의들

정의3:  $v$ 를  $T_M$ 의 임의의 참노드라 하자.  $v$ 의 참자식들을 그것의 Depth가 작은 것부터 순서대로  $c_1, c_2, c_3, \dots$ 와 같이 놓고, 자연수  $l_i := depth(c_i) - depth(v)$ 라 하자 (2.2절의 정규언어군의 참노드는 항상 적어도 두 개 이상의 참자식을 가진다. 또  $\forall i \geq 2, x_i := \frac{l_1 + l_2 + \dots + l_i}{i-1}$ 라 하자.

방금 정의한  $x_i$ 에 대해서 우리는 다음의 성질을 이끌어 낼 수 있다.

정리1:  $x_2 > x_3 > \dots > x_d \leq x_{d+1} \leq \dots$ 를 만족하는 상수  $d$ 는 언제나 존재한다.  
( $x_2 \leq x_3$ 일때,  $d=2$ )

정의4: 정리1에서 정의한  $d$ 와  $x_d$ 에 대해서,  $x_d$ 를 넘지 않는 최대의 정수를  $h_M$ 라 하고, 1과  $d-1$  사이의 정수  $\lambda_M := (d-1)(h_M+1) - (l_1 + \dots + l_d)$ 라 놓자.  
 $= (d-1)(h_M+1 - x_d)$

그림1-그림3에서 보여진 DFA에 대한  $l_i$ 들과  $d, h_M, \lambda_M$ 은 다음과 같다.

$$M_1: x_2 > x_3 = (2+3+3)/2 = 4$$

$$= (2+3+3+4)/3 = x_4;$$

$$d = 3; h_{M_1} = 4; \lambda_{M_1} = 2$$

$$M_2: x_2 = 2 + 3 = 5; d = 2; h_{M_2} = 5; \lambda_{M_2} = 1$$

$$M_3: x_2 > \dots > x_5 = (3+4+5+5+5)/4 = 5.5 < x_6;$$

$$d = 5; h_{M_3} = 5; \lambda_{M_3} = 2$$

정의5: 주어진 DFA  $M$ 과 트리  $T_M$ 이 있다.  $T_M$ 의 루트는 참노드로 간주한다.  $V_t$ 는  $depth(u) \leq t$ 인 참노드  $u$ 들의 집합이고,  $W_m$ 은  $T_M$ 의 참노드들을 위에서부터 즉 Depth가 작은것부터 나열했을 때,  $m$ 번째 참노드까지의 집합이다 주어진 참노드의 집합  $V$ 에 대해서,  $gl(V)$ 는 참부모가  $V$ 에 속하나 자신은  $V$ 에 속하지 않는 참노드들의 집합이고,  $gl_k(V)$ 는  $gl(V)$ 의 참노드들을 Depth가 작은것부터 나열했을 때  $k$ 번째 참노드까지의 집합이다.

#### 3.2 결과

이 절에서는 우리의 결과를 설명한다 이를 위해서는 III.1에서 정의한 개념들 외에도 정리2에서 정의될 두 개의 수열  $\{L_m\}, \{R_m\}$ 의 도움이 필요하다. 우리의 결과는 정리2와 정리3의 내용이다

정의 6: 주어진 수  $m$ 에 대하여,  $v_m$ 은  $T_M$ 의  $m$ 번째 참노드라 할 때, 두 수  $L_m$ 과  $R_m$ 은 다음과 같이 정의된다

\*  $x_d$ 가 정수, 즉  $x_d = h_M$ 인 경우:

$$L_m := |\{u \in gl(W_m) : depth(u) \leq depth(v_m)\}|$$

$$R_m := |\{u \in gl(W_m) : depth(u) \leq depth(v_m) + d - 2\}|$$

\*  $x_d$ 가 정수가 아닌 경우, 즉  $x_d > h_M$ :

$$L_m := |\{u \in gl(W_m) : depth(u) \leq depth(v_m) - \lambda_M\}|$$

$$R_m := |\{u \in gl(W_m) : depth(u) \leq depth(v_m) + d - 2 - \lambda_M\}|$$

정리2: 모든 자연수  $m$ 에 대하여,  $L_m < R_m + 1 = L_{m+1}$ 을 만족한다.

정리2에 의하면,  $L_1$ 보다 큰 임의의 수  $n$ 에 대해서,  $n \in [L_m, R_m]$ 인 구간 혹은 숫자  $m$ 이 유일하

게, 언제나 존재한다. 뿐만 아니라, 상수  $h_M$ 이나  $\lambda_M$ , 그리고 정의6의 두 수  $L_m, R_m$ 은 오직 DFA  $M$ 에만 의존하는 수들이다 따라서, 주어진 DFA  $M$ 에 대해서, 우리는 구간들  $\{[L_m, R_m]\}$ 을 계산할 수 있다. 일례로, 그림1에 제시된 DFA  $M_1$ 에 대해,  $L_1 = 12, R_1 = 21, L_2 = 22, R_2 = 23, L_3 = 24, R_3 = 40$ 임을 그림5를 통해 확인해 볼 수 있다. 이와 같이, 구간  $[L_m, R_m]$ 과 구간  $[L_{m+1}, R_{m+1}]$ 은 서로 겹치지 않으며 연달아 있다. 따라서,  $L_1$ 보다 큰 임의의 수  $n$ 에 대해서,  $n \in [L_m, R_m]$ 인 구간 혹은 숫자  $m$ 이 유일하게, 언제나 존재한다. 위의 예에서  $n = 30$ 에 대해서,  $30 \in [L_3, R_3]$ 이므로,  $m = 3$ 이 된다. 그러면, 이제 정리3을 살펴보자.

정리3:  $L_m \leq n \leq R_m$ 인 모든  $n$ 에 대하여, 트리  $T_n^m := W_m \cup gl_n(W_m)$ 은  $n$ 개의 **leaf**를 갖는 트리 중 최소 외부경로길이 혹은 최소 Cost를 갖는다.

결국, 정리2와 정리3에 의하면, 주어진 DFA  $M$ 과 자연수  $n$ 에 대해서, 우리는 문제2에서 정의한 최적트리  $n$ 개의 **leaf**를 갖는  $T_M$ 의 부분트리 중 최소 외부경로길이 혹은 최소 Cost를 갖는 트리를 찾을 수 있다. 우선 (i) 모든  $m$ 에 대해  $L_m$ 과  $R_m$ 을 계산한 후, (ii) 주어진  $n$ 에 대해서  $n \in [L_m, R_m]$ 을 만족하는  $m$ 을 찾는다. 그러면,  $W_m$ 을 참내부노드 집합으로 하고, 이로부터 뺀어 나온 최상위  $n$ 개의 **leaf**들을 갖는 트리가 바로 최적트리인 것이다. 앞의 예를 따르면, 주어진 DFA  $M_1$ 과 임의의 숫자  $n$ 에 대해서, 우리는  $n$ 개의 **leaf**를 갖는 부분트리 중 최소 외부경로길이 혹은 Cost를 갖는 트리를 찾으려 한다. 그러면, (i) 주어진 DFA  $M_1$ 으로부터 구간들의 열  $\{[L_m, R_m]\}$ 을 구한 다음, (ii) 수  $n$ 에 대해서  $n \in [L_m, R_m]$ 을 만족하는  $m$ 을 찾는다. 예를 들어,  $n = 30$ 에 대해서,  $m = 3$ 을 구한다. 그러면,  $W_3$ 을 참내부노드 집합으로 하고, 이로부터 뺀어 나온 최상위 30개의 **leaf**들을 갖는 트리가 바로 30개의 **leaf**를 갖는 트리중의 최적트리인 것이고, 이는 문제1-Prefix coding 문제로 해석해 보면, 패턴 ‘01’로 끝나는 30개의 단어로 이루어진 Prefix 코드 중에서 단어길이의 합이 가장 작은 코드를 얻

게 된 것이다.

### 3.3 정리2와 정리3의 증명

정리2와 정리3의 엄밀한 증명은 매우 복잡하고 길다. 특히,  $L_m$ 과  $R_m$ 의 정의 자체가 DFA  $M$ 에 의해 결정되는 상수  $x_d$ 가 정수인가 아닌가에 따라 두 경우로 나뉘어 있기 때문에, 증명의 본질적인 아이디어는 같음에도 불구하고 두 경우로 나뉘어 증명되어야 한다. 따라서, 여기에서는 ‘ $x_d$ 가 정수인 경우, 즉  $x_d = h_M$ 인 경우’에 대해서만 증명하기로 하자.

#### 3.3.1 필요한 보조정리들

이제부터  $T$ 는 언제나  $n$ 개의 **leaf**를 갖는 최적트리 ( $n$ 개의 **leaf**를 갖는 트리들 중 최소 외부경로길이를 갖는 것)이고,  $W_T$ 는  $T$ 의 참내부노드들의 집합,  $\kappa_T := \max_{\{v \in W_T\}} \text{depth}(v)$ 라 하자. 또,  $n$ 개의 **leaf**를 갖는 최적트리들의 참내부노드 개수의 최대값을  $f_M(n)$ 이라 하자. 즉,  $f_M(n) := \max\{|W_T| : T \text{는 } n \text{개의 } \text{leaf} \text{를 가진 최적트리}\}$ 이다. 보조정리1은  $d$ 와  $h_M$ 의 정의로부터 보조정리2는  $T$ 가 최적트리라는 사실로부터 쉽게 유추될 수 있다.

보조정리1: (i)  $l_d < x_d \leq l_{d+1}$  이다.  
 (ii)  $v$ 는  $T_M$ 의 임의의 참노드이고,  $c_i$ 는  $v$ 의  $i$ 번째 참자식이라 하자. 그러면,  
 $\text{depth}(c_d) \leq \text{depth}(v) + h_M - 1$  이고,  
 $\text{depth}(c_{d+1}) \geq \text{depth}(v) + h_M$  을 만족한다.

보조정리2:  $u$ 와  $v$ 는  $\text{depth}(u) < \text{depth}(v)$ 를 만족하는  $T_M$ 의 임의의 참노드이다. 최적트리  $T$ 는 항상 다음을 만족한다.  
 (i)  $v \in W_T$ 이면,  $u \in W_T$   
 (ii)  $u, v \in gl(W_T)$ 이고  $v \in T$ 이면,  $u \in T$

이제 좀더 복잡한 보조정리들을 소개할까 한다. 먼저 소개된 보조정리1과 보조정리2는 정리2와 정리3의 증명에 요긴하게 쓰이고, 앞으로 소개될 두 보조정리도 정리3의 증명에 필수적이다.

보조정리3: 최적트리  $T$ 는 항상  $\text{depth}(u) \leq \kappa_T + h_M - 1$ 인  $u \in gl(W_T)$ 를 포함해야 한다.

보조정리4:  $T$ 는  $n$ 개의 참leaf와  $f_M(n)$ 개의 참내부노드를 가진 최적트리이다  $v$ 는  $T$ 의 참leaf 중 가장 Depth가 작은 것이다 그러면,  
 $|\{u \in gl(W_T) \cap T : depth(u) \geq depth(v) + h_M\}| \leq d - 2.$

3.3.2 정리2의 증명

$L_m < R_m + 1$ 은  $L_m$ 과  $R_m$ 의 정의를 보면 자명하다. 그러므로,  $R_m + 1 = L_{m+1}$ 만을 증명하면 된다. 정의에 의하면  $L_{m+1}$ 은 참부모가  $W_{m+1}$  ( $T_M$ 에서 위로부터  $m + 1$ 번째 참노드까지 모은 집합)에 속하면서, Depth가  $depth(v_{m+1}) + h_M - 1$  이하인 참노드들의 개수이다 그러면, 이 집합은, 참부모가  $v_{m+1}$ 인 노드들과 참부모가  $W_m$ 에 속하는 것들로 나뉜다. 보조정리1에 의하면, 전자의 개수는  $d$ 이다. 반면에, 후자는

$$\{u \in gl(W_m) : depth(u) \leq depth(v_{m+1}) + h_M - 1, u \neq v_{m+1}\}$$

이어서 그 개수가  $(R_m - (d - 2) - 1)$ 이다. 따라서, 우리는  $L_{m+1} = d + (R_m - d + 1) = R_m + 1$ 을 얻는다.

3.3.3 정리3의 증명

주어진 수  $n$ 에 대해,  $m$ 은  $L_m \leq n \leq R_m$ 을 만족하는 수라 하자.  $f_M(n)$ 의 정의에 의해서  $n$ 개의 참leaf와  $f_M(n)$ 개의 참내부노드를 갖는 최적트리는 존재한다. 그리고, 보조정리2에 의해 이 최적트리는  $T_n^{f_M(n)}$ 이어야 함을 알 수 있다 즉,  $T_n^{f_M(n)}$ 은  $n$ 개의 참leaf를 갖는 최적트리이다. 반면, 우리의 목적은  $T_n^m$ 이  $n$ 개의 참leaf를 갖는 최적트리임을 보이는 것이다. 우리는,  $m$ 이 아닌 어떠한  $\gamma$ 에 대해서도  $T_n^\gamma$ 가  $T_n^{f_M(n)}$ 일 수 없음을 보임으로써,  $T_n^m = T_n^{f_M(n)}$ 이고 따라서 최적트리라는 것을 보일 것이다.  $\gamma \leq m - 1$ 인 경우와  $\gamma \geq m + 1$ 인 경우로 나누어서 증명한다.

\*  $\gamma \leq m - 1$ 인 경우: 이 경우, 우리는  $T_n^\gamma$ 가  $depth(u) \geq depth(v_{\gamma+1}) + h_M$ 인 참leaf  $u$ 를  $d - 1$ 개 이상 갖는다는 것을 보인다  $T_n^\gamma$ 에서  $v_{\gamma+1}$

은 Depth가 가장 작은 참leaf이므로, 보조정리4에 의하여,  $T_n^\gamma$ 은  $T_n^{f_M(n)}$ 과 같을 수 없음이 증명된다 먼저  $\gamma = m - 1$ 인 경우를 보자.  $T_n^{m-1}$ 은  $T_n^m$ 에서 참내부노드  $v_m$ 를 참자식들을 모두 떼어내 참leaf로 변화시키고, 잃은 만큼의 참leaf를  $gl(W_{m-1}) - gl_n(W_m)$ 에서 가져다 붙인 것이다(그림 6). 우리는 이때, (i) 떼어낸  $v_m$ 의 참자식들이 적어도  $d$ 개, 따라서  $gl(W_{m-1}) - gl_n(W_m)$ 에서 가져다 붙여야 하는 참leaf가  $v_m$ 을 제외하고 적어도  $d - 1$ 개인 것과, (ii) 이들 모두 Depth가 적어도  $depth(v_m) + h_M$ 임을 보인다. 그러면,  $T_n^{m-1}$ 는  $depth(u) \geq depth(v_m) + h_M$ 인 참leaf  $u$ 를  $d - 1$ 개 이상 갖게 되어  $\gamma = m - 1$ 에 대한 증명이 끝난다.  $n$ 은  $L_m$ 과  $R_m$ 사이의 수이므로, 정의6으로부터  $gl_n(W_m)$ 은  $W_m$ 의 참자식이며 Depth가  $depth(v_m) + h_M - 1$  이하인 모든 참노드를 갖고 있음을 알 수 있다. 고로,  $v_m$ 의 참자식 중 Depth가  $depth(v_m) + h_M - 1$  이하인 모든 참노드가 이에 포함되므로, 보조정리1에 의하여, 이들의 수는 적어도  $d$ 인 것을 알 수 있어 (i)이 증명된다. (ii)를 증명하기 위해,  $gl(W_{m-1}) - gl_n(W_m)$ 을 살펴보자. 참부모가  $W_{m-1}$ 에 포함되는 참노드는  $v_m$ 을 제외하고 모두  $gl(W_m)$ 에 포함된다. 만일 그것의 Depth가  $depth(v_m) + h_M - 1$ 이었다면, 앞에서 살펴본 바와 같이  $gl_n(W_m)$ 에 포함되었을 것이므로,  $gl(W_{m-1}) - gl_n(W_m)$ 에 포함되어 있는 참노드는 모두 Depth가 적어도  $depth(v_m) + h_M$ 이다. 고로 (ii)가 증명되었다. 이와 같은 방법으로  $\gamma \leq m - 2$ 인 모든 경우에 대해 단계적으로 참내부노드  $v_{\gamma+1}$ 을 참leaf로 바꾸고, 대신  $gl(W_\gamma) - gl_n(W_{\gamma+1})$ 로부터 참노드를 가져다 붙이는 식으로  $T_n^\gamma$ 을 만들 수 있는데, 참leaf의 개수는  $n$ 으로 일정해야 하는 반면, 이를 만들 수 있는 참부모의 개수는 갈수록 줄어드므로, 지워지는  $v_{\gamma+1}$ 의 참자식 수는  $d$  이상으로 계속 늘어나고, 대신 붙여지는 참노드의 depth는 갈수록  $depth(v_{\gamma+1}) + h_M$ 보다 커진다. 고로,  $T_n^\gamma$ 가  $depth(u) \geq depth(v_{\gamma+1}) + h_M$ 인 참leaf  $u$ 를  $d - 1$ 개 이상 가질 수밖에 없음이 더욱 자명해진다

\*  $\gamma \geq m + 1$ 인 경우: 이 경우, 우리는  $depth(u) \leq depth(v_\gamma) + h_M - 1$ 이고,  $gl(W_\gamma)$ 에 포함되면서  $gl_n(W_\gamma)$ 에 포함되지 못한 참노드  $u$ 가 있음을 보일

것이다. 그러면,  $T_n^\gamma$ 에서  $depth(v_\gamma) = k_{T_n^\gamma}$ 를 만족하므로, 보조정리3에 의하여,  $T_n^\gamma$ 은 최적트리일 수 없음이 증명된다. 먼저  $\gamma = m + 1$ 인 경우를 보자.  $T_n^{m+1}$ 은  $T_n^m$ 에서  $\text{leaf } v_{m+1}$ 을 참자식들을 붙여 참내부노드로 변화시키고, 붙인 만큼의  $\text{leaf}$ 를  $gl_n(W_m)$ 에서 제거한 것이다 (그림6). 한편,  $n$ 이  $L_m$ 과  $R_m$ 사이의 수이므로 정의6으로부터  $gl_n(W_m)$ 은  $W_m$ 의 참자식이면서 Depth가  $depth(v_{m+1}) + h_M$  이상인  $\text{leaf}$ 를 기껏해야  $d - 2$ 개 갖고 있음을 알 수 있다. 따라서, 우리가 붙여야 하는  $v_{m+1}$ 의 참자식들이 적어도  $d$ 개, 따라서  $gl_n(W_m)$ 에서 제거해야 하는  $\text{leaf}$ 가  $v_{m+1}$ 을 제외하고 적어도  $d - 1$ 개임을 보이면, Depth가  $depth(v_{m+1}) + h_M - 1$  이하이고,  $gl(W_{m+1})$ 에 포함되면서  $gl_n(W_{m+1})$ 에 포함되지 못한 참노드  $u$ 가 있음을 보여 증명을 끝낸 셈이 된다. 그러면,  $T_n^m$ 으로부터  $T_n^{m+1}$ 으로 가기 위해  $v_{m+1}$ 에 붙여야 하는  $v_{m+1}$ 의 참자식들이 적어도  $d$ 개임을 보이자 우선 Depth가  $depth(v_{m+1}) + h_M - 1$  이하의 참자식들은 모두 붙여야 한다 그렇지 않으면, 붙여지지 않은 이 노드 자체가 위의 노드  $u$ 의 조건을 다 만족하여 증명이 끝난다. 그러면,  $v_{m+1}$ 의 참자식이면서 Depth가  $depth(v_{m+1}) + h_M - 1$  이하인 참노드는 모두 몇 개인가 보조정리1에 의하면  $d$ 개이다. 고로,  $\gamma = m + 1$ 인 경우에 대해 증명이 끝났다. 동일한 방법으로,  $\gamma \geq m + 2$ 인 모든 경우에 대해 단계적으로  $\text{leaf } v_\gamma$ 을 참내부노드로 바꾸고, 대신 붙인 만큼의  $\text{leaf}$ 를  $gl_n(W_{\gamma-1})$ 에서 제거한다.  $\text{leaf}$ 의 개수는  $n$ 으로 일정해야 하는 반면, 이를 만들 수 있는 참부모의 개수는 갈수록 늘어나므로, 제거되는  $gl_n(W_{\gamma-1})$ 의  $\text{leaf}$ 의 Depth는 갈수록  $depth(v_\gamma) + h_M - 1$ 보다 작아진다는 것을 알 수 있다. 고로,  $depth(u) \leq depth(v_\gamma) + h_M - 1$ 이고  $gl(W_\gamma)$ 에 포함되면서,  $gl_n(W_\gamma)$ 에 포함되지 못한 참노드  $u$ 가 있을 수밖에 없음은 더욱 자명해진다.

#### IV. 결론

이상으로 우리는 II.2절에서 소개된 정규언어군에 속하는 임의의 정규언어  $L$ 에 대해서, 그 안의 최적 Prefix 코드  $C_n \subseteq L$ 을 찾는 방법을 보였다. 이는,

$L$ 에 대응하는 DFA  $M$ 과 자연수  $n$ 에 대해서, 문제에서 정의한 최적트리  $n$ 개의  $\text{leaf}$ 를 갖는  $T_M$ 의 부분트리 중 최소 외부경로길이 혹은 최소 Cost를 갖는 트리를 찾음에 의해서였다. (i) 모든  $m$ 에 대해  $L_m$ 과  $R_m$ 을 계산한 후, (ii) 주어진  $n$ 에 대해서  $n \in [L_m, R_m]$ 을 만족하는  $m$ 을 찾는다. 그러면,  $W_m$ 을 참내부노드 집합으로 하고, 이로부터 뺀어 나온 최상위  $n$ 개의  $\text{leaf}$ 들을 갖는 트리가 최적트리이고, 이 트리의  $\text{leaf}$ 들에 대응하는 단어들의 집합이 바로 우리가 찾는 최적 Prefix 코드  $C_n \subseteq L$ 이다. 관련된 주제로서 앞으로 연구해 볼 만한 흥미로운 문제는, II.2절에서 소개한 정규언어군 외에도 좀더 일반적인 정규언어군에 대해 최적 Prefix 코드를 찾는 방법은 무엇인가이다.

#### 참고 문헌

- [1] 정옥현, 윤영석, 호요성. "Huffman 부호와 평균 부호길이 함수의 특성을 이용한 양방향 가변 길이 부호의 생성 방법, *Proceedings of the IEK Conference 2003*, p.137-140, 2003.
- [2] T. Berger, R. W. Yeung. "Optimum "1"-ended Binary Prefix Codes", *IEEE Transactions on Information Theory* 36(3), p. 1435-1441, 1990.
- [3] R. M. Capocelli, A. De Santis, G. Persiano. "Binary Prefix Codes Ending in a "1"", *IEEE Transactions on Information Theory*, 40(1), p. 1296-1301, 1994.
- [4] Chan Sze-Lok, M. Golin. "A Dynamic Programming Algorithm for Constructing Optimal "1"-ended Binary Prefix-Free Codes", *IEEE Transactions on Information Theory*, 46(4), p. 1637-1644, 2000.
- [5] T. H. Cormen, C. E. Leiserson, R. L. Rivest. *Introduction to Algorithms*, MIT Press, 1990.
- [6] V. S. Choi, M. J. Golin. "Lopsided Trees I: Analyses", *Algorithmica*, 31, p.240-290, 2001.
- [7] S. Kapoor, E. Reingold. "Optimum Lopsided Binary Trees", *Journal of the Association for Computing Machinery*, 36(3), p.573-590. 1989.
- [8] S. A. Savari. "Some Notes on Varn Coding", *IEEE Transactions on Information Theory*, 40(1), p. 181-186, 1994.
- [9] S. A. Savari. "A Probabilistic Approach to Some

Asymptotics in Noiseless Communications”, *IEEE Transactions on Information Theory*, 46(4), p. 1246-1262, 2000.

- [10] B. F. Varn. “Optimal Variable Length Codes (Arbitrary Symbol Costs and Equal Code Word Probabilities)”, *Informat. Contr.* 19, p. 289-301, 1971.

<부록 그림>

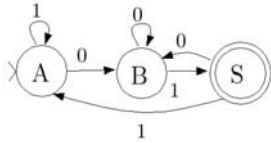
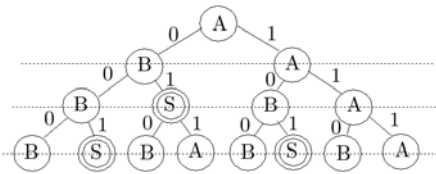


그림 1. (a) '01'로 끝나는 단어를 받아들이는 DFA  $M_1$



(b) DFA  $M_1$ 의 무한트리  $T_{M_1}$ ; State S를 갖고 있는 노드들이 참노드이다.

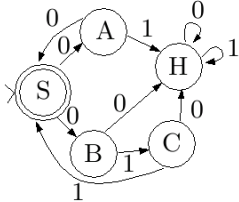


그림 2. (a)  $(00 + 111)^*$ 를 받아들이는 DFA  $M_2$

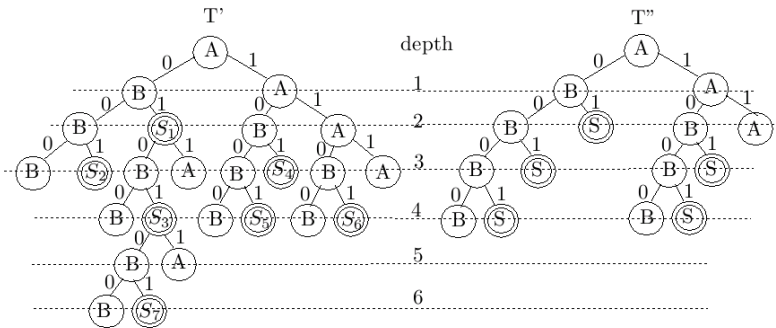


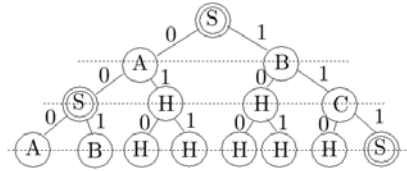
그림 4. 5개의 참leaf를 갖는 두 부분트리  $T', T'' \subseteq T_{M_2}$ ;

$cost(T') = 3 + 3 + 4 + 4 + 6 = 20$  이고

$cost(T'') = 2 + 3 + 3 + 4 + 4 = 16$  이다.

$T'$ 에 대응하는 코드는  $\{001, 101, 1001, 1101, 010101\}$  이고

$T''$ 에 대응하는 코드는  $\{01, 001, 101, 0001, 1001\}$  이다.



(b) DFA  $M_2$ 의 무한트리  $T_{M_2}$ ; State S를 갖고 있는 노드들이 참노드, H는 참자식을 갖지 않는다

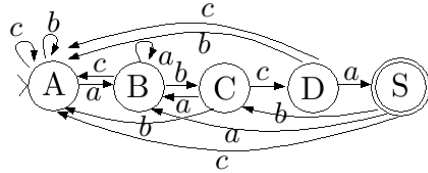
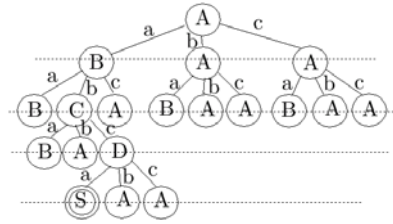


그림 3. (a) 'abca'로 끝나는 단어를 받아들이는 DFA  $M_3$



(b) DFA  $M_3$ 의 무한트리  $T_{M_3}$ ; State S를 갖고 있는 노드들이 참노드이다.



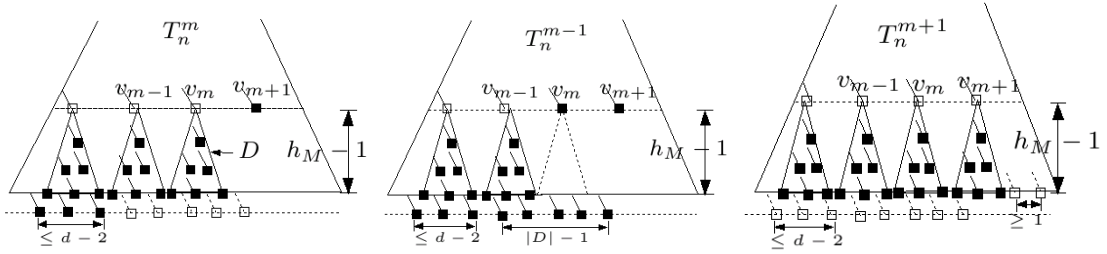


그림 5. 정리3의 증명 과정:  $\gamma = m - 1$  인 경우와  $\gamma = m + 1$  인 경우;  $T_n^m$  으로부터  $T_n^{m-1}$ ,  $T_n^m$  으로부터  $T_n^{m+1}$  을 만드는 과정을 관찰하라.

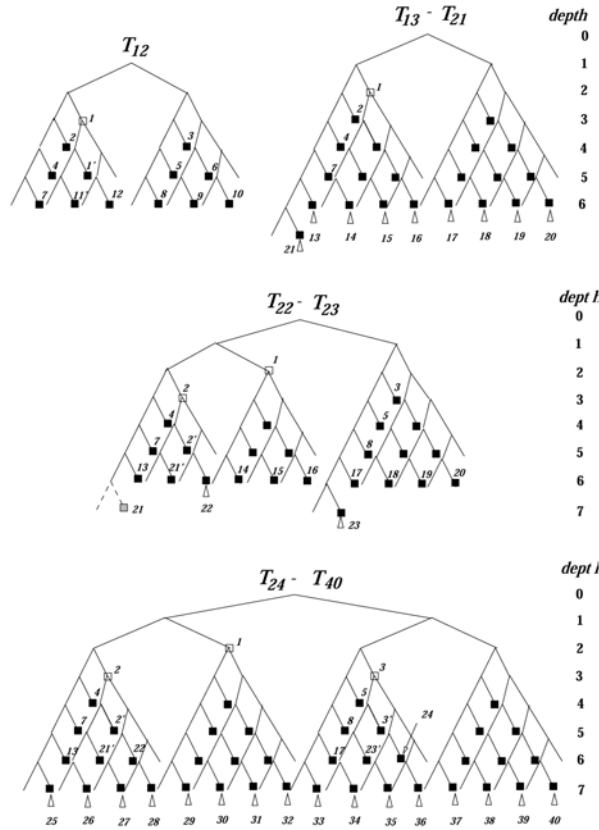


그림 6. 그림1의 DFA  $M_1$  ('01'로 끝나는 단어를 받아들이는 DFA)에 대한 최적 Prefix 트리열  $T_{12} \sim T_{40}$  과, 그의 Construction 과정.

나 현 숙(Hyeon-Suk Na)

정회원

1993년 2월 서울대학교 수학과 졸업

1995년 2월 포항공대 수학과 석사

2002년 2월 포항공대 수학과 박사