Horner's Method 와 Chien Search 의 효율성 측정에 관한 연구

인재휘, 김동찬
국민대학교 정보보안암호수학과

{jhin0303, dckim}@kookmin.ac.kr

A Study on the Efficiency Measurement of Horner's Method and Chien Search

JaeHui In, Dong-Chan Kim
Kookmin Univ.

요 약

BCH 부호의 디코딩은 5 단계로 구성된다. 디코딩할 때 수신된 벡터의 신드롬 값과 오류 벡터의 해밍무게, 위치정보, 크기정보를 계산해야 한다. 이 중 오류벡터의 위치정보를 계산하는 과정에서 유한체 상에서 정의한 다항식의 근을 구해야 한다. 다항식의 근을 구하는 방식으로는 Horner's Method, Chien Search 등이 존재한다. 이 중 Chien Search 알고리듬이 더 낮은 계산 복잡도를 가진다. 본 논문에서는 BCH 부호의 정의와 성질, Chien Search 알고리듬에 대해 소개하고 Sage 로 구현하여 효율성을 알아본다.

Ⅰ. 서 론

BCH 부호는 신드롬 값과 오류벡터의 해밍무게, 위치정보, 크기정보를 계산하여 오류벡터를 구하고 이를 통해 부호어를 복원한다. 이 때 오류벡터의 해밍무게는 Peterson-Gorenstein-Zierler 알고리듬과 Berlekamp-Massy 알고리듬으로, 크기정보는 Forney 알고리듬으로 계산 가능하다.

오류 벡터의 위치정보는 유한체 상에서 정의한 다항식의 근을 구하는 과정을 통해 얻을 수 있다. 이 과정은 Horner's Method, Chien Search 알고리듬 등을 이용하여 계산 가능하다. 일반적인 전수조사 방식은 $O(t^2)$ 곱셈 연산과 $O(t)$ 덧셈 연산이 필요하며 Horner's method 는 $O(t)$ 곱셈 연산과 $O(t)$ 덧셈 연산이 필요하다. Chien Search 알고리듬은 $O(t)$ 상수 곱셈 연산과 $O(t)$ 덧셈 연산으로 더 효율적으로 근을 구할 수 있다.

본 논문에서는 BCH 부호와 다항식의 근을 효율적으로 구하는 Chien Search 에 대해 소개한다. Ⅱ 절에서는 기호 및 정의에 대해 소개한다. Ⅲ 절에서는 BCH 부호를 정의하고 BCH 부호의 최소거리에 대한 정리를 증명한다. Ⅳ 절에서는 Chien Search 알고리듬을 소개하고 Ⅴ 절에서는 Horner's Method 와 Chien Search 알고리듬의 시간 비교 결과를 나타낸다.

Ⅱ. 기호 및 정의

본 논문에서는 다음의 기호 및 정의를 사용한다.

q	소수 p의 거듭제곱
m	$q^m \equiv 1 (mod\, n)$ 을 만족하는 가장 작은 정수
F_q	원소의 개수가 q개인 유한체
α	F_{q^m} 에서 위수가 n인 원소
δ	설계된 거리, $2 \le \delta \le n$
$\mathcal{H}$	선형부호 $\mathcal{C} = [n,k]_q$의 패리티검사행렬
$F(x)$	F_q상에서 정의된 t차 다항식

Ⅲ. BCH 부호

정의. 임의의 양의 정수 s에 대해 길이가 n인 BCH 부호 $\mathcal{C}$ 는 다음의 패리티검사행렬 $\mathcal{H}$ 로 정의하는 부호이다.

$$\mathcal{H} = \begin{bmatrix} 1 & \alpha^{s} & \alpha^{2s} & \cdots & \alpha^{(n-1)s} \\ 1 & \alpha^{s+1} & \alpha^{2(s+1)} & \cdots & \alpha^{(n-1)(s+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{s+\delta-2} & \alpha^{2(s+\delta-2)} & \cdots & \alpha^{(n-1)(s+\delta-2)} \end{bmatrix}.$$

이 때 $s = 1$이면 Narrow-sense BCH 부호, $n = q^m - 1$이면 Primitive BCH 부호라 하며, 두 조건 모두 만족할 때 Primitive Narrow-sense BCH 부호라 한다.

BCH 부호의 최소거리는 다음의 정리를 통해 구할 수 있다.

정리. BCH 부호 $\mathcal{C}$의 패리티검사행렬의 행의 개수가 $\delta - 1$일 때, $\mathcal{C}$의 최소거리는 δ보다 크거나 같다.

증명. 부호 $\mathcal{C}$의 패리티검사행렬의 임의의 $\delta - 1$개의 열이 일차 독립일 때, $\mathcal{C}$의 최소거리는 δ보다 크다.

$\mathcal{C}$ 의 패리티검사행렬에서 $b_\gamma (1 \le \gamma \le \delta - 1)$ 번째 열을 선택하여 생성한 행렬의 행렬식은 다음과 같다.

$$det \begin{bmatrix} \alpha^{b_1 s} & \alpha^{b_2 s} & \cdots & \alpha^{b_{\delta-1} s} \\ \alpha^{b_1 (s+1)} & \alpha^{b_2 (s+1)} & \cdots & \alpha^{b_{\delta-1} (s+1)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{b_1 (s+\delta-2)} & \alpha^{b_2 (s+\delta-2)} & \cdots & \alpha^{b_{\delta-1} (s+\delta-2)} \end{bmatrix} =$$

$$(\alpha^{b_1 s}\, \alpha^{b_2 s} \cdots \alpha^{b_{\delta-1} s}) det \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha^{b_1} & \alpha^{b_2} & \cdots & \alpha^{b_{\delta-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{b_1 (\delta-2)} & \alpha^{b_2 (\delta-2)} & \cdots & \alpha^{b_{\delta-1} (\delta-2)} \end{bmatrix}.$$

서로 다른 α^{b_γ} 은 0 이 아니고 우변의 행렬은 $Vandermonde$ 행렬이기에 0 이 될 수 없다. 그러므로

BCH 부호 C 의 패리티검사행렬에서 임의로 뽑은 $\delta-1$개의 열은 일차독립을 만족한다. ■

IV. Chien Search 알고리듬

Chien Search 알고리듬은 F_q 상에서 정의한 다항식 $F(x)$ 의 근을 찾는 알고리듬이다. Chein Search 알고리듬은 F_q 의 원시원인 ω 에 대해 다음의 식을 이용하여 근을 찾을 수 있다.

$$F(\omega^i) = f_0 + f_1(\omega^i) + f_2(\omega^i)^2 + \cdots + f_t(\omega^i)^t.$$

$$F(\omega^{i+1}) = f_0 + f_1(\omega^{i+1}) + f_2(\omega^{i+1})^2 + \cdots + f_t(\omega^{i+1})^t$$
$$= f_0 + f_1(\omega^i)\omega + f_2(\omega^i)^2\omega^2 + \cdots + f_t(\omega^i)^t\omega^t.$$

위의 식에서 $F(\omega^i)$의 각 항을 $\beta_{j,i}(0 \le j \le t)$로 표현하면 다음과 같다.

$$F(\omega^i) = \beta_{0,i} + \beta_{1,i} + \beta_{2,i} + \cdots + \beta_{t,i} = \sum_{j=0}^{t} \beta_{j,i}.$$

$\beta_{j,i+1}(0 \le j \le t)$를 $\beta_{j,i}\omega^j$라 하면 $F(\omega^{i+1})$를 다음과 같이 표현할 수 있다.

$$F(\omega^{i+1}) = \beta_{0,i} + \beta_{1,i}\omega + \beta_{2,i}\omega^2 + \cdots + \beta_{t,i}\omega^t$$
$$= \beta_{0,i+1} + \beta_{1,i+1} + \beta_{2,i+1} + \cdots + \beta_{t,i+1} = \sum_{j=0}^{t} \beta_{j,i+1}.$$

이를 통해 $F(\omega^{i-1})$의 각 항에 ω^j를 곱하는 것으로 계산 복잡도를 줄일 수 있다.

ω는 F_q의 원시원이기에 $\omega^i(0 \le i \le q-1)$로 F_q상의 0이 아닌 모든 원소를 표현할 수 있다. 그러므로 모든 $F(\omega^i)$를 계산하는 것은 F_q의 모든 원소를 대입해보는 것과 동일하다. 이 때 사전계산 테이블 $T_j(0 \le j \le t)$를 이용한다. T_j에는 f_j와 F_q 상의 모든 원소를 곱한 결과를 저장한다. 이 방식의 수행과정은 알고리듬 1 과 같다.

알고리듬 1: Chien Search 알고리듬

입력 : $F(X)(= f_0 + f_1X + \cdots + f_tX^t)$,
사전계산 테이블 $T_j\ (1 \le j \le t)$

출력 : $rootset$

```
1:  rootset ← ∅
2:  For i = 1 to t do
3:      β_i ← f_i
4:  end for
5:  If f_0 = 0 then
6:      rootset ← rootset ∪ {0}
7:  end if
8:  For i = 1 to q − 1 do
9:      For j = 1 to t do
10:         β_j ← T_j[β_j]
11:     end for
12:     If Σ_{k=0}^{t} β_k = 0 then
13:         rootset ← rootset ∪ {ω^i}
14:     end if
15: end for
16: return rootset
```

V. 구현 결과

실험 환경은 다음과 같다.

-하드웨어 환경 2.30GHz 더블 코어 Intel Core-i5, 4GB RAM

-프로그래밍 환경 SageMath 9.1

측정에 사용한 알고리듬은 Horner's Method 와 Chien Search 알고리듬이며 HQC 에서 제안한 파라미터에 대해 이루어졌다. 파라미터별로 측정한 연산 시간은 [표 1]과 같다. 시간 측정 단위는 밀리 초(ms)이다.

[표 1] 파라미터별 알고리듬의 측정결과

파라미터			알고리듬	시간(ms)
n	q	m		
1023	2	10	Horner's Method	626.674
			Chien Search	157.695
255	2	8	Horner's Method	21.008
			Chien Search	3.092

결과적으로 Chien Search 알고리듬이 Horner's Method 에 비해 최대 6.7 배 빠른 결과를 보였다.

VI. 결론

본 논문에서는 BCH 부호의 정의와 최소거리, Chien Search 알고리듬에 대해 소개하고 SAGE 로 구현한 Horner's Method 와 Chien Search 알고리듬의 속도를 비교하였다. 결과적으로 Chien Search 알고리듬의 연산시간이 최대 6.7 배 더 빠른 것을 보였다. 이를 통해 Horner's Method 보다 Chien Search 알고리듬이 속도 측면에서 더 효율적임을 알 수 있었다.

추후에는 C 언어로 두 알고리듬을 구현하여 속도 측정 및 효율성 비교를 진행할 예정이다.

참 고 문 헌

[1] R. C. Bose, and D. K. Ray-Chaudhuri, "On A Class of Error Correcting Binary Group Codes," Information and Control, 3 (1): 68-79, March 1960.

[2] Junho Cho and Wonyong Sung, "Software implementation of Chien search process for strong BCH codes," 2008 IEEE International Symposium on Circuits and Systems, 2008, pp. 1842-1845.

[3] A. Hocquenghem, "Codes correcteurs d'erreurs," Chiffres (in French), Paris, 2: 147-156, September 1959.

[4] W. C. Huffman and V. Pless, "Fundamentals of Error-Correcting Codes," Cambridge, MA: University Press, 2003.

[5] Melchor, C.A., et al.: Hamming quasi-cyclic (HQC). Technical report, National Institute of Standards and Technology 2017 (2018)