1 ความรู้เบื้องต้น
Basic Concepts

1.1 บทนำ

ทฤษฎีรหัสเป็นวิชาที่เพิ่งเกิดขึ้นใหม่ มีจุดเริ่มต้นในปี ค.ศ. 1948 เมื่อนักคณิตศาสตร์และคอมพิวเตอร์ชาวอเมริกัน ชื่อ Claude Elwood Shannon ได้เผยแพร่ผลงานของเขาเรื่อง A Mathematical Theory of Communication ในวารสาร Bell System Technical Journal ในบทความนี้ เขาได้เสนอทฤษฎีที่สำคัญมากต่อการสื่อสาร เราไม่อาจพูดถึงทฤษฎีนี้อย่างถูกต้อง โดยไม่ให้คำจำกัดความของคำบางคำเพิ่มเติม แต่เราอาจกล่าวถึงทฤษฎีนี้ได้โดยคร่าว ๆ ว่า

> ตราบใดที่อัตราของรหัสมีค่าน้อยกว่าจำนวน ๆ หนึ่งที่เรียกว่า ความสามารถของช่องสัญญาณ ซึ่งเป็นตัววัดปริมาณของสารสนเทศที่ช่องสัญญาณสามารถส่งได้ เป็นไปได้เสมอที่จะส่งสารสนเทศที่มีข้อผิดพลาดน้อย โดยการส่งรหัสที่มีความยาวพอสมควร

จากบทความนี้ ทำให้นักคณิตศาสตร์เริ่มหันมาสนใจศึกษาค้นคว้าเกี่ยวกับเรื่องรหัสกันอย่างกว้างขวาง ซึ่งเป็นจุดเริ่มต้นของวิชาสองวิชาคือ วิชา ***ทฤษฎีสารสนเทศ*** (Information Theory) และวิชา ***ทฤษฎีรหัส*** (Coding Theory) ทั้งสองวิชานี้มีวัตถุประสงค์ที่จะทำให้การสื่อสารมีประสิทธิภาพและความน่าเชื่อถือ ในสภาวะที่มีสิ่งรบกวน ***การสื่อสารที่มีประสิทธิภาพ*** (efficient) คือการสื่อสารที่ใช้เวลาน้อย ส่วน ***การสื่อสารที่มีความน่าเชื่อถือ*** (reliable) คือการสื่อสารที่ข้อมูลที่ส่งเหมือนหรือใกล้เคียงกับข้อมูลที่รับมากที่สุดเท่าที่จะเป็นได้ วิชาทฤษฎีสารสนเทศ จะเน้นศึกษาเกี่ยวกับประสิทธิภาพในการสื่อสาร ซึ่งมีเนื้อ

หาเน้นไปทางเรื่องความน่าจะเป็นและการวิเคราะห์ ส่วนวิชาทฤษฎีรหัสจะเน้นศึกษาเกี่ยวกับการสร้างรหัสที่ดี โดยใช้ความรู้พื้นฐานทางพีชคณิต รหัสที่ดีในที่นี้หมายถึง

1. รหัสที่เสียเวลาในการส่งน้อย
2. รหัสที่มีความถูกต้องแม่นยำสูง สามารถตรวจจับและแก้ไขข้อผิดพลาดได้
3. มีขั้นตอนการเข้ารหัส-ถอดรหัสที่มีประสิทธิภาพ

รหัสแก้ไขข้อผิดพลาด (error correcting code) เป็นแขนงหนึ่งของวิชาทฤษฎีรหัส ที่ศึกษาเกี่ยวกับวิธีสร้างรหัส สำหรับใช้ในการสื่อสาร ที่มีความสามารถในการตรวจจับและแก้ไขข้อผิดพลาดของสารสนเทศได้

คำว่า "สารสนเทศ" เป็นศัพท์บัญญัติทางคอมพิวเตอร์ ซึ่งมาจากคำภาษาอังกฤษว่า "information" ราชบัณฑิตยสถานได้ให้ความหมายของสารสนเทศไว้ในพจนานุกรมว่าหมายถึง ข่าวสาร; การแสดงหรือชี้แจงข่าวสารข้อมูลต่าง ๆ ดังนั้น ในที่นี้จะใช้คำว่าสารสนเทศและข่าวสารสลับกันได้

เชื่อว่าผู้อ่านคงจะได้คุ้นเคยกับรหัส ที่ใช้ในชีวิตประจำวันกันบ้างแล้ว เช่น รหัสมอร์สที่ใช้ในการสื่อสารทางโทรเลข รหัสรีด-มูลเลอร์ที่ใช้ในการสื่อสารระหว่างยานอวกาศ ชื่อมารีนเนอร์กับสถานีบนพื้นโลก หรือรหัส ISBN ที่ใช้ในการระบุบนหนังสือแต่ละเล่ม

รหัสรีด-มูลเลอร์ (Reed-Muller Code)

ผู้อ่านคงจะเคยเห็นภาพถ่ายของดาวอังคาร ดาวเสาร์ และดาวเคราะห์ดวงอื่น ๆ กันบ้างแล้ว ในปี ค.ศ. 1965 ยานมารีนเนอร์ 4 (Mariners 4) เป็นยานอวกาศลำแรกที่ส่งภาพของดาวอังคารมายังพื้นโลก ในการส่งภาพแต่ละภาพ กระทำโดยแบ่งภาพที่จะส่งออกเป็นตารางเล็ก ๆ และระบุความเข้มของตารางเล็ก ๆ แต่ละตารางเหล่านี้ ด้วยลำดับของเลข 0 และ 1 ที่มีความยาว 6 ตำแหน่ง หรือ 6 บิต (bit ย่อมา

จาก binary digit) ซึ่งจะเรียกลำดับของ 0 และ 1 นี้ว่าสาร ดังนั้นจะมีสารที่เป็นไปได้ทั้งหมด 64 สาร ซึ่งจะแทนความเข้มได้ถึง 64 ระดับ ตั้งแต่ 000000 ซึ่งแทนสีขาวที่มีความเข้มน้อยที่สุด ไปจนถึง 111111 ซึ่งแทนสีดำที่มีความเข้มมากที่สุด เมื่อสารถูกส่งมาถึงพื้นโลก สารที่ได้รับอาจจะไม่ตรงกับสารที่ส่ง อาจทำให้ได้รับภาพที่ผิดเพี้ยนไป ดังนั้น จึงจำเป็นต้องมีการป้องกันข้อผิดพลาดโดยการเข้ารหัสสาร โดยการทำให้สารยาวขึ้นด้วยการเพิ่มตำแหน่งบางตำแหน่งเข้าไปในสาร เช่น รหัสที่ใช้ในยานมารีนเนอร์ 9 เป็นรหัสที่คำรหัสแต่ละคำมีความยาว 32 บิต รหัสดังกล่าวนี้คือรหัสรหัสรีด-มูลเลอร์-(32,64,16) ซึ่งเพิ่มความยาวของสารจากเดิม 6 บิตเป็น 32 บิต

รหัสมอร์ส(Morse Code)

รหัสในยุคแรก ๆที่เราอาจได้เคยพบเห็นในชีวิตประจำวัน ได้แก่ รหัสมอร์ส (morse code) ที่ใช้ในการส่งโทรเลข ข่าวสารที่ส่ง จะอยู่ในรูปของตัวอักษร อักษรแต่ละตัว จะถูกแทนด้วยลำดับของจุดและขีด เราเรียกลำดับเหล่านี้ว่าคำรหัส (code word) คำรหัสเหล่านี้จะถูกส่งไปตามสายโทรเลข สู่ปลายทาง

Morse Code

A	._	N	_.
B	_...	O	___
C	_._.	P	.__.
D	_..	Q	__._
E	.	R	._.
F	.._.	S	...
G	__.	T	_
H	....	U	.._
I	..	V	..._
J	.___	W	.__
K	_._	X	_.._
L	._..	Y	_.__
M	__	Z	__..

พนักงานที่อยู่ปลายทาง จะแปลลำดับของจุดและขีดที่ได้รับกลับเป็นตัวอักษร จุดและขีดบางตัวอาจผิดไปจากที่ส่ง เพราะถูกรบกวนในระหว่างเดินทางมาตามสายโทรเลข ซึ่งอาจทำให้การแปลความหมายผิดไปได้

รหัส ISBN (ISBN Code)

ISBN ย่อมาจาก International Standard Book Number ISBN เป็นรหัสฐาน 11 ที่มีความยาว 10 หลัก แต่ละหลักเป็นเลข 0, 1, 2, ..., 9 และ X ตัวอย่างเช่น 0-19-859617-0 เครื่องหมาย - ไม่ใช่ส่วนหนึ่งของคำรหัส แต่มีไว้เพื่อให้อ่านรหัสได้ง่ายเท่านั้น 0 ในตำแหน่งแรกบอกให้รู้ว่าหนังสือเล่มนี้เป็นภาษาอังกฤษ 19 ในสองตำแหน่งถัดมาแทนสำนักพิมพ์ Oxford University Press หกตำแหน่งถัดมาคือ 859617 เป็นหมายเลขของหนังสือเล่มนั้นซึ่งกำหนดโดยสำนักพิมพ์ ส่วน 0 ในตำแหน่งสุดท้ายจะเป็นเลขที่เลือกให้สอดคล้องกับสมการ

$$\sum_{i=1}^{10} ic_i \equiv 0 \pmod{11}$$

เมื่อ c_i เป็นตำแหน่งแต่ละตำแหน่งในคำรหัส ISBN อีกตัวอย่างหนึ่งคือ 0550 – 10206 – X ซึ่งเป็น ISBN ของหนังสือชื่อ Chambers Twentieth Century Dictionary อักษร X ในที่นี้แทนเลข 10

เนื้อหาและทฤษฎีที่เกี่ยวข้องกับรหัสแก้ไขข้อผิดพลาด ที่จะกล่าวถึงในหนังสือเล่มนี้ สามารถนำไปประยุกต์ใช้กับสถานการณ์ทั้งหลายที่มีลักษณะร่วมกันดังนี้ คือ มีข่าวสารจากแหล่งใดแหล่งหนึ่งที่ต้องการจะส่งผ่านช่องสัญญาณไปยังผู้รับปลายทาง เช่น การสนทนาทางโทรศัพท์ การส่งโทรเลข การส่งภาพถ่ายจากยานอวกาศมายังพื้นโลก หรือการเก็บบันทึกข้อมูลบนเทปแม่เหล็กหรือแผ่นซีดี เป็นต้น

1.2 รหัสต้นทางและรหัสช่องสัญญาณ (Source Code and Channel Code)

รหัสแบ่งออกเป็น 2 ประเภท คือ

1. **รหัสต้นทาง** (source code)

สมมุติว่าแหล่งกำเนิดต้นทางมีสารสนเทศ ซึ่งเป็นข้อความที่จะสื่อสารกับผู้รับปลายทาง 2 ข้อความ คือข้อความ 'ถอย' และ 'บุก' เรา

อาจแทน 'ถอย' ด้วยเลข 0 และแทน 'บุก' ด้วยเลข 1 ดังในตาราง 1.2.1

ตาราง 1.2.1

ข้อความ	รหัสต้นทาง
ถอย	0
บุก	1

เราเรียกการแทนข้อความด้วย 0 และ 1 ว่า ***การเข้ารหัสต้นทาง*** (source encoding) และเรียก 0 และ 1 ว่า ***รหัสต้นทาง*** แต่ถ้าเรามีข้อความมากขึ้น เช่นสมมุติว่ามีข้อความ 4 ข้อความ ดังปรากฏในหลักแรกของตาราง 1.2.2

ตาราง 1.2.2

ข้อความ	รหัสต้นทาง
จู่โจมจากทางทิศเหนือ	00
จู่โจมจากทางทิศตะวันตก	01
จู่โจมจากทางทิศตะวันออก	10
จู่โจมจากทางทิศใต้	11

แหล่งกำเนิดต้นทางอาจแทนข้อความทั้งสี่นั้นด้วย 00, 01, 10 และ 11 ตามลำดับ ดังนั้น 00, 01, 10, 11 เป็นรหัสต้นทาง ในกรณีทั่วไป เราอาจคิดถึงรหัสต้นทางในรูปลำดับของสัญลักษณ์ จากเซตจำกัดของพยัญชนะ ซึ่งส่วนใหญ่มักจะใช้ {0,1} เป็นเซตของพยัญชนะ ดังนั้นรหัสต้นทางก็คือลำดับของ 0 และ 1 นั่นเอง

2. รหัสช่องสัญญาณ (channel code)

รหัสช่องสัญญาณ คือรหัสที่จะช่วยให้เราสามารถตรวจจับข้อผิดพลาดได้ และถ้ารหัสนั้นดีพอก็จะช่วยให้เราสามารถแก้ไขข้อผิดพลาดได้อีกด้วย เพื่อความสะดวก เราจะเรียกรหัสต้นทางว่า ***สาร*** (message) ถ้าเราส่งสารผ่านช่องสัญญาณที่มีสิ่งรบกวน บางตำแหน่ง

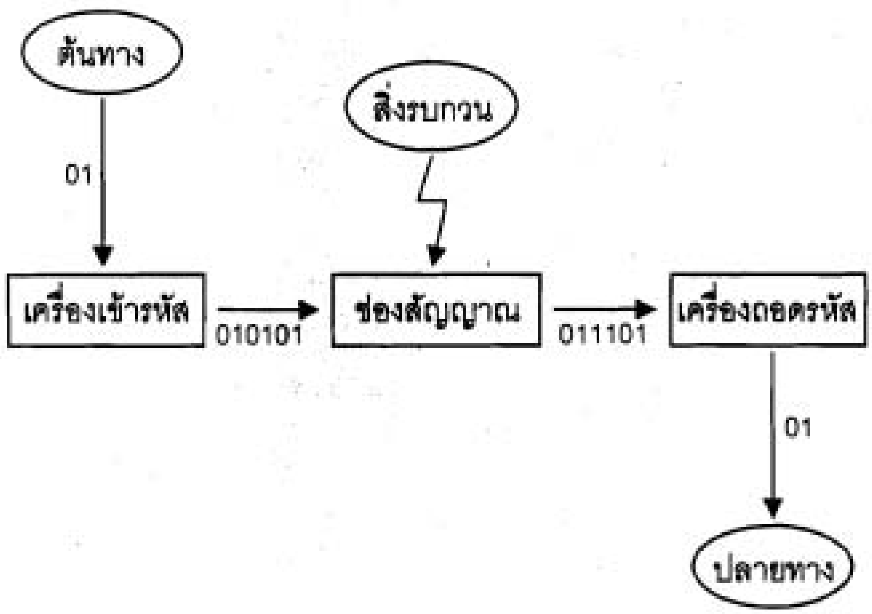
ของสารอาจถูกเปลี่ยนไป เพื่อป้องกันข้อผิดพลาดนี้ เราจะเพิ่มบางตำแหน่งเข้าไปในสารก่อนที่จะส่งสาร เรียกกระบวนการนี้ว่าการเข้ารหัส และเรียกผลลัพธ์ที่ได้จากการเพิ่มบางตำแหน่งเข้าไปในสารนี้ว่า ***รหัสช่องสัญญาณ*** การเข้ารหัสสารจะทำให้สารยาวขึ้น ส่วนที่ยาวขึ้นนี้จะช่วยในการตรวจจับหรืออาจแก้ไขข้อผิดพลาดได้ ตลอดหนังสือเล่มนี้ เราจะเน้นศึกษาเฉพาะรหัสช่องสัญญาณนี้เท่านั้น

1.3 ระบบสื่อสาร (Communication System)

เราจะไม่เน้นศึกษาการเข้ารหัสต้นทาง ดังนั้น เราจะเริ่มต้นจากสมมุติว่าเรามีรหัสต้นทางอยู่แล้ว ซึ่งเราจะเรียกรหัสต้นทางนี้ว่าสาร สารจะถูกส่งจากต้นทาง ผ่านช่องสัญญาณไปยังผู้รับปลาย ทาง เช่น การสื่อสารทางโทรศัพท์ จากที่แห่งหนึ่งไปยังที่อีกแห่งหนึ่ง ช่องสัญญาณในการสื่อสารทางโทรศัพท์ ได้แก่สายโทรศัพท์ การส่งสารจากดาวเทียมหรือยานอวกาศ สู่สถานีบนพื้นโลก มีชั้นบรรยากาศนอกโลก พร้อมกับอุปกรณ์สำหรับรับส่งข่าวสารเป็นช่องสัญญาณ การบันทึกข้อมูลหรือเพลงบนเทปหรือแผ่นซีดี ถือเป็นการสื่อสารจากปัจจุบันไปสู่อนาคต ในกรณีนี้ ช่องสัญญาณก็คือ เทปหรือแผ่นซีดีนั่นเอง

จะเห็นว่าไม่มีช่องสัญญาณใดที่ดีสมบูรณ์แบบ ช่องสัญญาณทั้งหลาย จะมีสิ่งที่อาจทำให้สารที่ส่งจากต้นทางเปลี่ยนแปลงไป เรียกสิ่งที่ทำให้สารเปลี่ยนแปลงไปว่า ***สิ่งรบกวน*** (noise) ดังนั้น สิ่งรบกวนก็คือสิ่งที่เป็นต้นเหตุให้ข่าวสารหรือข้อมูลที่ได้รับ แตกต่างไปจากข่าวสารที่ส่ง ซึ่งอาจจะเป็น ฟ้าแลบ ฟ้าร้อง อุณหภูมิ หรือ รอยขีดข่วนบนเทปหรือแผ่นซีดีก็ได้ ช่องสัญญาณที่น่าเชื่อถือได้ ต้องเป็นช่องสัญญาณซึ่งเมื่อสารถูกส่งจากต้นทาง และถ้ามีบางตำแหน่งถูกเปลี่ยนไปแล้ว ผู้รับปลายทางสามารถแก้ไขให้ถูกต้องได้ หรืออย่างน้อยก็รู้ว่ามีข้อผิดพลาดเกิดขึ้น และอาจขอให้ต้นทางส่งสารมาใหม่(retransmit)

ถ้าเป็นไปได้ เช่น การพูดคุยทางโทรศัพท์ ถ้าปลายทางได้ยินไม่ชัดเจน สามารถขอให้ต้นทางพูดใหม่ได้ ส่วนการสื่อสารในอวกาศ เป็นไปไม่ได้เลยที่จะให้ต้นทางส่งสารมาใหม่ เพราะจะเสียเวลาและค่าใช้จ่ายมาก เพื่อป้องกันข้อผิดพลาด จะต้องมีการเข้ารหัสสารก่อนที่จะส่งผ่านช่องสัญญาณ ดังนั้นจึงต้องมีเครื่องเข้ารหัส สารจะถูกส่งไปยังเครื่องเข้ารหัส เพื่อแปลงสารให้เป็นคำรหัส แล้วจึงค่อยส่งคำรหัสผ่านช่องสัญญาณที่มีสิ่งรบกวน คำที่ผ่านช่องสัญญาณจะถูกส่งไปยังเครื่องถอดรหัส เพื่อตัดสินว่าสารที่ส่งจากต้นทางคืออะไร แล้วจึงค่อยส่งผลต่อไปยังปลายทาง ดังแสดงในรูป 1.3.1



รูป 1.3.1 : แบบจำลองของระบบสื่อสารทั่วไป

ในรูป 1.3.1 สมมุติว่าสารที่จะส่ง คือ 01 สารนี้จะถูกส่งเข้าไปยังเครื่องเข้ารหัส เครื่องเข้ารหัสจะเปลี่ยนสาร 01 ให้เป็นคำรหัส 010101 เมื่อคำรหัสถูกส่งผ่านช่องสัญญาณที่มีสิ่งรบกวน ตำแหน่งบางตำแหน่งของคำรหัสอาจถูกเปลี่ยนไป เช่น อาจจะเปลี่ยนไปเป็น 011101 นั่นคือมีข้อผิดพลาดเกิดขึ้นในตำแหน่งที่สาม เมื่อเครื่องถอดรหัสได้รับคำ

011101 ที่ผ่านมาตามช่องสัญญาณ เครื่องถอดรหัสจะตัดสินจากคำที่ได้รับนี้ว่าคำที่ส่งคือคำใด จะเห็นว่า คำรหัสในที่นี้ก็คือลำดับของเลข 0 หรือ 1 เช่นกัน

1.4 เครื่องเข้ารหัส (Encoder)

เครื่องเข้ารหัสเป็นส่วนหนึ่งของเครื่องมือสื่อสาร ที่ทำหน้าที่แปลงสารให้เป็นคำรหัส เพื่อเพิ่มความน่าเชื่อถือให้แก่ช่องสัญญาณหรือเครื่องมือสื่อสาร ถ้ามีข้อผิดพลาดระหว่างการส่งข้อมูล ผู้รับปลายทางควรต้องรู้ว่ามีข้อผิดพลาดเกิดขึ้น และอาจแก้ไขให้ถูกต้องได้ เครื่องเข้ารหัสจะทำหน้าที่แปลงสารให้เป็นคำรหัส โดยการเพิ่มตำแหน่งบางตำแหน่งเข้าไปในสาร ทำให้สารยาวขึ้น ในทางคอมพิวเตอร์ จะเรียกส่วนที่เพิ่มเข้าไปในสารนี้ว่า ***ส่วนซ้ำซ้อน*** ซึ่งตรงกับภาษาอังกฤษว่า redundancy ศัพท์คำนี้อาจทำให้ผู้อ่านเข้าใจผิดได้ว่าส่วนที่เพิ่มเข้าไปในสารต้องซ้ำกับสาร จริง ๆ แล้วการเพิ่มส่วนนี้มีขั้นตอนวิธีที่แตกต่างกัน ส่วนที่เพิ่มไม่จำเป็นต้องซ้ำกับสารเสมอ ในที่นี้จะเรียกส่วนนี้ว่า ***ส่วนตรวจสอบ***

สมมุติว่ามีสาร 0 แทน 'บุก' และ 1 แทน 'ถอย' และสมมุติว่ามีนายทหารที่ต้องการส่งสารให้หน่วยทหารของตน ให้ถอยจากการรบ นายทหารท่านนี้จึงส่ง 1 ผ่านช่องสัญญาณที่มีสิ่งรบกวนไปยังหน่วยทหาร สมมุติว่า 1 ถูกเปลี่ยนไปเป็น 0 หน่วยทหารที่ได้รับ 0 จะไม่มีโอกาสรู้เลยว่ามีข้อผิดพลาดเกิดขึ้น เขาอาจเข้าใจผิด คิดว่านายทหารที่เป็นผู้บังคับบัญชาต้องการให้บุก ซึ่งอาจทำให้เกิดความเสียหายตามมา ปัญหานี้เกิดจากไม่มีการเข้ารหัสสารก่อนที่จะส่ง การเข้ารหัสสารทำได้หลากหลายวิธี ขึ้นอยู่กับว่าเราต้องการให้รหัสที่ได้มีประสิทธิภาพและความน่าเชื่อถือมากน้อยเพียงใด

ตัวอย่าง 1.4.1 : สมมุติว่าเราเข้ารหัสสารโดยการเพิ่มตำแหน่งที่ซ้ำกันเข้าไปในสารอีกหนึ่งตำแหน่ง ดังนั้น 1 กลายเป็น 11 ซึ่งแทนข้อความ 'ถอย' และ 0 จะกลายเป็น 00 ซึ่งแทนข้อความ 'บุก' ในกรณีนี้ เราเรียก 11 และ 00 ว่า ***คำรหัส*** ดังในตาราง 1.4.1 และเรียกกระบวนการแปลง 1 และ 0 ให้เป็น 11 และ 00 ตามลำดับว่าการ ***เข้ารหัส***

ตาราง 1.4.1

สาร	คำรหัส
0	00
1	11

ไม่ว่าต้นทางจะส่ง 11 หรือ 00 ถ้ามีข้อผิดพลาดเกิดขึ้นหนึ่งตำแหน่ง ผู้รับปลายทางจะได้รับ 01 หรือ 10 ผู้รับปลายทางจะรู้ได้ทันทีว่ามีข้อผิดพลาดเกิดขึ้น เพราะทั้ง 01 และ 10 ไม่ใช่คำรหัส แต่ก็ไม่รู้ว่าเกิดข้อผิดพลาดที่ตำแหน่งใด

ตาราง 1.4.2

สาร	คำรหัส
0	000
1	111

นายทหารท่านนี้สามารถปรับปรุงรหัสของเขาโดยการส่งสารซ้ำสามครั้ง นั่นคือส่ง 111 แทน 1 และส่ง 000 แทน 0 จะได้รหัสดังในตาราง 1.4.2

จะเห็นว่ารหัสในตาราง 1.4.2 ดีกว่ารหัสในตาราง 1.4.1 ทั้งนี้เพราะว่า ถ้ามีข้อผิดพลาดเกิดขึ้นหนึ่งหรือสองตำแหน่ง คำที่ได้รับปลายทางจะไม่ใช่คำรหัส ทำให้ผู้รับปลายทางรู้ได้ว่ามีข้อผิดพลาดเกิดขึ้น นอกจากนี้ ถ้าเรารู้เพิ่มเติมว่า ช่องสัญญาณไม่มีโอกาสทำให้คำรหัสที่ส่งผิดไปสองตำแหน่งแล้ว ผู้รับปลายทางนอกจากจะรู้ว่ามีข้อผิดพลาดเกิดขึ้นแล้วยังรู้อีกว่ามีข้อผิดพลาดเกิดขึ้นที่ใด และสามารถแก้ไข

ให้ถูกต้องได้ เช่น สมมุติว่าปลายทางได้รับ 010 ผู้รับจะรู้ว่ามีข้อผิดพลาดเกิดขึ้น เพราะ 010 ไม่ใช่คำรหัส และรู้ว่าผิดในตำแหน่งที่สอง นั่นคือรู้ว่าคำรหัสที่ส่งจากต้นทางต้องเป็น 000 จะเป็น 111 ไม่ได้ เพราะนั่นต้องแสดงว่า มีข้อผิดพลาดเกิดขึ้นสองตำแหน่งคือตำแหน่งที่หนึ่งและสาม เรียกรหัสในตาราง 1.4.1 และ 1.4.2 นี้ว่า ***รหัสแบบซ้ำ*** (repetition code) ที่มีความยาวเท่ากับ 2 และ 3 ตามลำดับ

ตัวอย่าง 1.4.2 : สมมุติว่าเรามีสาร 00, 01, 10 และ 11 ถ้าเราส่งสารเหล่านี้โดยไม่มีการเข้ารหัส เช่น สมมุติว่าต้นทางต้องการจะส่ง 10 ซึ่งแทนข้อความ "จู่โจมจากทางทิศตะวันออก" เมื่อรหัสผ่านช่องสัญญาณที่มีสิ่งรบกวน 10 อาจจะถูกเปลี่ยนเป็น 11 เมื่อผู้รับปลายทางรับสาร 11 ก็จะเข้าใจว่าต้นทางต้องการให้จู่โจมจากทางทิศใต้ จึงจำเป็นต้องมีการเข้ารหัสก่อนที่จะส่งข่าวสารนั้นผ่านช่องสัญญาณที่มีสิ่งรบกวน

เราอาจเข้ารหัสสารโดยการเพิ่มสารให้มีความยาวเพิ่มขึ้นอีกหนึ่งตำแหน่ง โดยจะเพิ่ม 0 หรือ 1 เข้าไปในแต่ละสาร เพื่อทำให้สารแต่ละสารมี 3 ตำแหน่ง และมีจำนวนเลข 1 เป็นจำนวนคู่ เช่น สาร 10 ก็จะถูกแปลงเป็น 101 ซึ่งมีสามหลักและจำนวนเลข 1 ใน 101 เท่ากับสอง ซึ่งเป็นจำนวนคู่ ส่วนสารอื่น ๆ ก็จะถูกแปลงหรือถูกเข้ารหัสในทำนองเดียวกัน ดังนั้น เราจะได้คำรหัสที่สมนัยกับสาร ดังปรากฏในตาราง 1.4.3 เรียกผลลัพธ์นี้ว่า **คำรหัส** และเรียก 0 หรือ 1 ที่เพิ่มเข้าไปนี้ว่า ***บิตตรวจสอบภาวะเสมอคู่*** (even parity check bit) สำหรับกรณีนี้ ถ้าเลข 0 และ 1 ที่เพิ่มเข้าไปแล้วทำให้จำนวนเลข 1 ในผลลัพธ์เป็นจำนวนคี่ เราจะเรียกบิตที่เพิ่มเข้าไปว่า ***บิตตรวจสอบภาวะเสมอคี่***(odd parity check bit) ในหนังสือเล่มนี้ เราจะพูดถึงเฉพาะบิตตรวจสอบภาวะเสมอคู่เท่านั้น และจะเรียกสั้น ๆ ว่า ***บิตตรวจสอบภาวะเสมอ*** (parity check bit)

ตาราง 1.4.3

สาร	คำรหัส
00	000
01	011
10	101
11	110

จะเห็นว่า สารแต่ละสารมีความยาวเท่ากันคือเท่ากับ 2 และคำรหัสแต่ละคำมีความยาวเท่ากันคือเท่ากับ 3 สัญลักษณ์ที่ใช้ในแต่ละตำแหน่งของคำรหัสมาจากเซต {0, 1}

ตัวอย่าง 1.4.3 : เราอาจเข้ารหัสสาร 00, 10, 01 และ 11 โดยการเพิ่มสารซ้ำเข้าไปอีกสองชุด เช่น เพิ่ม 01 เข้าไปในสาร 01 อีกสองชุด จะได้คำรหัส

ตาราง 1.4.4

สาร	คำรหัส
00	000000
10	101010
01	010101
11	111111

010101 และเข้ารหัสสารอื่น ๆ ในทำนองเดียวกัน จะได้ผลลัพธ์ดังปรากฏในตาราง 1.4.4 สารแต่ละสารมีความยาวเท่ากันคือเท่ากับ 2 และคำรหัสแต่ละคำมีความยาวเท่ากันคือเท่ากับ 6 สัญลักษณ์แต่ละตัวในคำรหัสมาจากเซต {0, 1}

1.5 ช่องสัญญาณ (Channel)

เราจะเน้นเฉพาะช่องสัญญาณที่เรียกว่า ***ช่องสัญญาณภินทนะที่ไม่มีหน่วยความจำ***(discrete memoryless channel หรือ DMC) เท่านั้น ช่องสัญญาณภินทนะหมายถึง ช่องสัญญาณที่มีข้อมูลเข้าและข้อมูลออก เป็นสัญลักษณ์ที่มาจากเซตจำกัด ช่องสัญญาณที่ไม่มีหน่วย

ความจำ คือช่องสัญญาณที่สัญลักษณ์แต่ละตัวที่ส่งเป็นอิสระต่อกัน การที่สัญลักษณ์ตัวหนึ่งจะผิด ไม่ขึ้นกับสัญลักษณ์ที่ส่งก่อนหน้าหรือหลังจากนั้น

สมมุติให้ $A = \{a_1, a_2, \ldots, a_q\}$ เป็นเซตของสัญลักษณ์ที่ใช้เป็นข้อมูลเข้าและข้อมูลออก บางครั้งช่องสัญญาณทำให้ข้อมูลเข้าไม่ตรงกับข้อมูลออก บางครั้งส่ง a_j แต่ปลายทางได้รับ a_i หรือบางครั้งช่องสัญญาณส่ง a_j ตัวเดิม แต่ปลายทางอาจได้รับ a_k ดังนั้น สัญลักษณ์แต่ละตัวใน A จะมีจำนวนจริง p ซึ่ง

$$P(\text{รับ } a_i \mid \text{ส่ง } a_j) = p \quad \text{สำหรับ } i \neq j$$

เป็นความน่าจะเป็นที่สัญลักษณ์ a_j ถูกส่งจากต้นทาง แต่ปลายทางได้รับ a_i ที่แตกต่างจากสัญลักษณ์ที่ส่ง เรียก p ว่า ***ความน่าจะเป็นไขว้*** (crossover probability) หรือ ***ความน่าจะเป็นที่สัญลักษณ์จะผิด*** (symbol error probability)

นิยาม 1.5.1

ช่องสัญญาณสำหรับสื่อสารประกอบด้วยเซต $A = \{a_1, a_2, \ldots, a_q\}$ ซึ่งเรียกว่าชุดตัวอักษรและเซตของความน่าจะเป็น $P(\text{รับ}a_i|\text{ส่ง}a_j)$ ที่สอดคล้องกับ

$$\sum_{i=1}^{q} P(\text{รับ } a_i \mid \text{ส่ง } a_j) = 1 \quad \text{สำหรับทุก ๆ } i$$

เมื่อ $P(\text{รับ } a_i \mid \text{ส่ง } a_j)$ มีค่าเท่ากัน สำหรับทุก ๆ a_j ใน A จะเรียกช่องสัญญาณนี้ว่า ช่องสัญญาณสมมาตรฐาน q (q-ary symmetric channel)

ดังนั้น ความน่าจะเป็นที่ a_j ถูกส่งจากต้นทาง แต่ปลายทางได้รับสัญลักษณ์ที่แตกต่างจาก a_j จะเท่ากับ

$$s = (q - 1)p$$

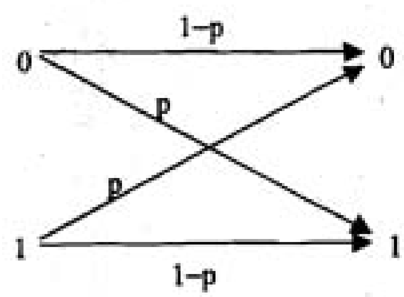
ดังนั้น ความน่าจะเป็นที่สัญลักษณ์ที่รับปลายทางตรงกับสัญลักษณ์ที่ส่งจากต้นทางเท่ากับ 1 – s นั่นคือ

$$P(\text{รับ } a_i \mid \text{ส่ง } a_i) = 1 - s = 1 - (q - 1)p$$

ในกรณีเฉพาะ เมื่อ q = 2 เราพิจารณาช่องสัญญาณสมมาตรฐานสอง (binary symmetric channel หรือ BSC) ซึ่งมี

$$P(\text{รับ } a_i \mid \text{ส่ง } a_j) = p$$

แล้ว ความน่าจะเป็นที่สัญลักษณ์ที่ได้รับตรงกับสัญลักษณ์ที่ส่งจากต้นทาง คือ $P(\text{รับ } a_j \mid \text{ส่ง } a_j) = 1 - p$ กล่าวคือ



รูป 1.5.1 : แผนภาพของช่องสัญญาณสมมาตรฐานสองหรือ BSC

$$P(\text{รับ } 1 \mid \text{ส่ง } 0) = P(\text{รับ } 0 \mid \text{ส่ง } 1) = p$$
$$P(\text{รับ } 0 \mid \text{ส่ง } 0) = P(\text{รับ } 1 \mid \text{ส่ง } 1) = 1 - p$$

เราสามารถอธิบายลักษณะของช่องสัญญาณ ด้วยแผนภาพเช่นช่องสัญญาณ BSC ที่แสดงในรูป 1.5.1 โดยปกติค่า p ของช่องสัญญาณ BSC จะน้อยกว่า $\frac{1}{2}$ เพราะถ้า p มีค่ามากกว่า $\frac{1}{2}$ ปลายทางจะเปลี่ยน 0 ให้เป็น 1 และเปลี่ยน 1 ให้เป็น 0

ข้อผิดพลาดที่เราจะเน้นในที่นี้ คือข้อผิดพลาดที่เรียกว่า ***ข้อผิดพลาดสุ่ม*** (random error) นั่นคือ ข้อผิดพลาดที่ตำแหน่ง

แต่ละตำแหน่งในลำดับมีโอกาสที่จะผิดเท่า ๆ กันและเป็นอิสระ ต่อกัน กล่าวคือ การที่ตำแหน่งใดตำแหน่งหนึ่งจะผิด ไม่ขึ้นอยู่กับข้อผิดพลาดของตำแหน่งอื่น ๆ

นิยาม 1.5.2 ถ้าแต่ละตำแหน่งในคำรหัส $c = c_1c_2 \ldots c_n$ มาจากชุดตัวอักษร A เราจะเรียก c ว่าคำรหัสบนชุดตัวอักษร A

ถ้าส่งคำรหัส $c = c_1c_2 \ldots c_n$ ที่มีความยาว n บนชุดตัวอักษร A โดยส่งสัญลักษณ์ทีละตัวผ่านช่องสัญญาณ BSC สมมุติว่าปลายทางได้รับ $x = x_1x_2 \ldots x_n$ แล้วความน่าจะเป็นที่จะได้รับ x เมื่อส่ง c จะเท่ากับ

$$P(\text{รับ } x \mid \text{ส่ง } c) = \prod_{i=1}^{n} P(\text{รับ } x_i \mid \text{ส่ง } c_i)$$

จะเห็นว่า ความน่าจะเป็นที่ปลายทางได้รับคำที่ไม่มีข้อผิดพลาดเลยจะเท่ากับ

$$P(\text{รับ } c \mid \text{ส่ง } c) = (1-p)^n$$

เพราะความน่าจะเป็นที่สัญลักษณ์ในแต่ละตำแหน่งจะถูกต้อง ตรงกับสัญลักษณ์ที่ส่ง มีค่าเท่ากับ $1 - p$ และคำที่ไม่มีข้อผิดพลาดเลยก็คือคำที่ถูกต้องทุกตำแหน่ง ถ้า x และ y แตกต่างกัน i ตำแหน่ง จะได้

$$P(\text{รับ } y \mid \text{ส่ง } x) = p^i(1-p)^{n-i}$$

ตัวอย่าง 1.5.1 : สมมุติให้ x = 100111 เป็นคำที่ส่งจากต้นทาง และ y = 110110 เป็นคำที่ปลายทางได้รับ จะเห็นว่าตำแหน่งที่ 2 และ 6 ของคำที่ได้รับผิดไปจากคำที่ส่ง ดังนั้น ความน่าจะเป็นที่จะได้รับ x เมื่อ y เป็นคำที่ส่งเท่ากับ

$$P(\text{รับ } y \mid \text{ส่ง } x) = p^2(1-p)^{6-2}$$

ถ้า p = 0.01 เราได้

$$P(\text{รับ } 110110 \mid \text{ส่ง } 100111) = (0.01)^2(1-0.01)^{6-2}$$
$$= (0.01)^2(0.99)^4 = 9.6059601 \times 10^{-5}$$

1.6 เครื่องถอดรหัส (Decoder)

เครื่องถอดรหัสจะทำหน้าแทนคำที่ได้รับปลายทาง ด้วยคำรหัส หรือไม่ก็ส่งสัญญาณว่ามีข้อผิดพลาดเกิดขึ้น เรียกการกระทำดังกล่าวนี้ว่า ***การถอดรหัส*** เครื่องถอดรหัสจะตัดสินจากคำที่ได้รับถ้าคำที่ได้รับไม่ใช่คำรหัส เครื่องถอดรหัสจะตรวจจับได้ว่าคำที่รับมานั้น มีข้อผิดพลาดเกิดขึ้น และถ้ารหัสดีพอ เครื่องถอดรหัสจะแก้ไขให้ถูกต้องได้ ถ้าผิดไม่มากนัก

ในตัวอย่าง 1.4.2 เรามีคำรหัส 4 คำคือ 000, 011, 101, 110 สมมุติว่าต้นทางส่งคำรหัส 101 ผ่านช่องสัญญาณที่มีสิ่งรบกวน สมมุติว่าคำรหัสผิดไปหนึ่งตำแหน่ง เครื่องถอดรหัสอาจจะได้รับ 001, 111 หรือ 100 ขึ้นอยู่กับข้อผิดพลาดว่าเกิดขึ้นที่ตำแหน่งใด ไม่ว่าเครื่องถอดรหัสจะได้รับ 001, 111 หรือ 100 เครื่องถอดรหัสจะรู้ได้ทันทีว่ามีข้อผิดพลาดเกิดขึ้น เพราะคำที่ได้รับ ไม่ว่าจะเป็น 001, 111 หรือ 100 ทั้งสามคำนี้ ไม่มีคำใดเป็นคำรหัสเลย

ในกรณีนี้ แสดงว่าเครื่องถอดรหัสตรวจจับได้ว่ามีข้อผิดพลาดเกิดขึ้น แต่ไม่รู้ว่าข้อผิดพลาดนั้นเกิดขึ้นที่ตำแหน่งใด ในตัวอย่าง 1.4.3 มีคำรหัส 4 คำเช่นกัน คือ 000000, 101010, 010101, 111111 ถ้าคำรหัส 010101 ถูกส่งผ่านช่องสัญญาณที่มีสิ่งรบกวน

สมมุติว่าเลข 0 ในตำแหน่งที่สามถูกเปลี่ยนไปเป็น 1 ดังนั้นคำรหัส 010101 จะถูกเปลี่ยนเป็น 011101 เครื่องถอดรหัสปลายทางจะรู้ทันทีว่ามีข้อผิดพลาดเกิดขึ้น เพราะ 011101 ไม่ตรงกับคำรหัสใดเลย นอกจากนี้เครื่องถอดรหัสยังสามารถตัดสินได้อีกว่า คำที่ส่งจากต้นทาง

ควรจะเป็น 010101 โดยใช้หลักเกณฑ์ที่เรียกว่า การถอดรหัสโดยใช้ความน่าจะเป็นสูงสุด ซึ่งจะได้ศึกษาในรายละเอียดต่อไป

1.7 รหัสและชุดตัวอักษร

เมื่อนึกถึงคำรหัส เราจะนึกถึงลำดับของสัญลักษณ์ ซึ่งเป็นสมาชิกของเซตจำกัดเซตใดเซตหนึ่ง และเมื่อกล่าวถึงรหัส เราจะหมายถึงเซตของคำรหัสทั้งหลาย

นิยาม 1.7.1 เรียก C ว่า **รหัสฐาน q** บนเซต $A = \{a_1, a_2, \ldots, a_q\}$ เมื่อ C คือเซตของลำดับของสัญลักษณ์ซึ่งมาจากเซตจำกัด A ที่มีสมาชิก q ตัว เรียกสมาชิกใน C ว่า ***คำรหัส*** และเรียกเซต A ว่า ***ชุดตัวอักษร*** (alphabet)

ถ้า q = 2 เราเรียก C ว่า ***รหัสฐานสอง*** หรือ ***รหัสไบนารี*** และ
ถ้า q = 3 เราเรียกรหัส C ว่า ***รหัสฐานสาม*** หรือ ***รหัสเทอร์นารี***

ตัวอย่าง 1.7.1 : ให้

$$C_1 = \{000, 111\}$$
$$C_2 = \{000, 011, 101, 110\}$$
$$C_3 = \{000000, 010101, 101010, 111111\}$$

จะเห็นว่า C_1, C_2, และ C_3 เป็นรหัสฐานสองหรือรหัสไบนารี เพราะตำแหน่งแต่ละตำแหน่งในคำรหัสเป็นสมาชิกของชุดตัวอักษร {0,1} ในกรณีนี้ เราอาจกล่าวว่า C_1, C_2, และ C_3 เป็นรหัสบนชุดตัวอักษร {0, 1}

ตัวอย่าง 1.7.2 : ให้ $C_4 = \{012210, 112112, 221020\}$

C_4 เป็นรหัสฐานสามหรือรหัสเทอร์นารี มีคำรหัส 3 คำ แต่ละคำมี 6 ตำแหน่ง แต่ละตำแหน่งมาจากชุดตัวอักษร {0, 1, 2}

ตัวอย่าง 1.7.3 : รหัส ISBN ที่เห็นแล้วในหัวข้อ 1.1 เป็นรหัสฐาน 11 แต่ละคำมี 10 หลัก และมี {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, X} เป็นชุดตัวอักษร ในที่นี้ X แทนเลข 10

หมายเหตุ :

1. สมาชิกในเซต A อาจจะเป็นตัวอักษร ตัวเลข หรือสัญลักษณ์ก็ได้ แต่เราจะเรียกรวมว่า สัญลักษณ์
2. ในหนังสือเล่มนี้เราศึกษารหัสบนฟิลด์จำกัด F_q ที่มีสมาชิก q ตัว (ดูรายละเอียดในบทที่ 2) แต่ตัวอย่างส่วนใหญ่จะเป็นรหัสบนฟิลด์ F_p เมื่อ p เป็นจำนวนเฉพาะ ในวิชาพีชคณิตนามธรรม เรารู้ว่าฟิลด์ F_p ไอโซมอร์ฟิกกับฟิลด์

$$\mathbf{Z}_p = \{0, 1, 2, \ldots, p-1\}$$

ดังนั้น เราจะใช้ $\mathbf{Z}_p$ แทนฟิลด์ F_p ยกเว้นเมื่อกล่าวเป็นอย่างอื่น

นิยาม 1.7.2

> เรียกจำนวนตำแหน่งในคำรหัสว่า ***ความยาว*** ของคำรหัส ถ้าคำรหัสใน C ทุกคำมีความยาวเท่ากัน เราจะเรียกเซต C ว่า ***รหัสแบบบล็อก*** (block code) และถ้า n คือความยาวของคำรหัส และ M คือจำนวนคำรหัสใน C จะเรียก C ว่า รหัส-(n, M)

ตัวอย่าง 1.7.4 :

1. ในตัวอย่าง 1.7.1 จะพบว่า C_1 เป็นรหัสไบนารี-(3, 2)
C_2 เป็นรหัสไบนารี-(3, 4)
C_3 เป็นรหัสไบนารี-(6, 4)
2. ในตัวอย่าง 1.7.2 จะพบว่า C_4 เป็นรหัสเทอร์นารี-(6, 3)

หมายเหตุ : รหัสมอร์สเป็นตัวอย่างของรหัสที่คำรหัสแต่ละคำมีความยาวไม่เท่ากัน ในที่นี้เราจะสนใจเฉพาะรหัสแบบบล็อกเท่านั้น ดังนั้น เมื่อกล่าวถึงรหัสเราจะหมายถึงรหัสแบบบล็อกนี้เท่านั้น ยกเว้นเมื่อระบุเป็นอย่างอื่น

สัญลักษณ์ : เราจะใช้ A^n แทนเซตของลำดับที่มีความยาว n ซึ่งแต่ละตำแหน่งในลำดับเป็นสมาชิกของ A

ตัวอย่าง 1.7.5 : ถ้า $F_2 = \{0, 1\}$ จะได้

$$F_2^3 = \{000, 100, 010, 001, 110, 101, 011, 111\}$$

จะเห็นว่าจำนวนสมาชิกใน F_2^3 เท่ากับ $2^3 = 8$ และรหัส C_1 และ C_2 ในตัวอย่าง 1.7.1 เป็นเซตย่อยของ F_2^3 ส่วนรหัส C_3 เป็นเซตย่อยของ F_2^6

ตัวอย่าง 1.7.6 : ถ้า q = 3 ให้ $F_3 = \{0, 1, 2\}$ เป็นชุดตัวอักษร เราได้

$$F_3^3 = \{ a_1a_2a_3 \mid a_i \in F_3 \text{ สำหรับ } i = 1, 2, 3\}$$

เราสามารถเลือก a_1, a_2 และ a_3 แต่ละตัวได้ 3 วิธี คือจะเลือกให้เป็น 0, 1 หรือ 2 ก็ได้ ดังนั้น จำนวนสมาชิกใน F_3^3 เท่ากับ $3^3 = 27$ หรือ

$$| F_3^3 | = 27$$

เมื่อ $| F_3^3 |$ แทนจำนวนสมาชิกในเซต F_3^3

หมายเหตุ : ในกรณีทั่วไป ถ้า $A = \{a_1, a_2, \ldots, a_q\}$ แล้ว จำนวนสมาชิกใน A^n จะเท่ากับ q^n หรือเขียน $| A^n | = q^n$

นอกจากรหัสแบบบล็อกแล้ว ยังมีรหัสอีกแบบหนึ่งที่ต้องใช้หน่วยความจำในการเข้ารหัส - ถอดรหัส สารจากต้นทางจะมีลักษณะเป็นบล็อก ๆ เช่นกัน แต่ละบล็อกจะถูกเข้ารหัสโดยขึ้นอยู่กับบล็อกบางบล็อกที่ส่งก่อนหน้านั้น เรียกรหัสประเภทนี้ว่า ***รหัสแบบต้นไม้*** (tree code) สาเหตุที่เรียกเช่นนี้เนื่องจากการเข้ารหัสแบบนี้สามารถอธิบายได้โดยง่ายด้วยกราฟต้นไม้

สมมุติฐานเบื้องต้น

เราจำเป็นต้องทำข้อตกลงเกี่ยวกับขอบเขตของสิ่งที่เราจะศึกษา ซึ่งเราจะเรียกข้อตกลงเหล่านี้ว่าข้อสมมุติฐาน ซึ่งได้แก่

1. เมื่อต้นทางส่งรหัสคำที่ยาว n ผู้รับปลายทางจะได้รับคำที่ยาว n เช่นกัน ถึงแม้จะมีบางตำแหน่งไม่ตรงกับที่ส่งจากต้นทาง จะไม่มีกรณีที่มีตำแหน่งบางตำแหน่งสูญหายไประหว่างการส่ง
2. โอกาสที่แต่ละตำแหน่งที่ส่งจะผิดพลาด จะเป็นอิสระต่อกัน การที่ตำแหน่งใดจะผิดพลาดไม่ขึ้นอยู่กับตำแหน่งข้างเคียง ซึ่งจะเรียกว่าข้อผิดพลาดแบบสุ่ม (random error)

1.8 หลักเกณฑ์การถอดรหัส (Decoding Rule)

ในระบบสื่อสารที่ต้องใช้รหัส คำที่ถูกส่งจากต้นทางต้องเป็นคำรหัสเท่านั้น ส่วนคำที่ได้รับปลายทางอาจจะเป็นคำรหัสหรือไม่ก็ได้ สมมุติว่า **x** เป็นคำที่ได้รับ ถ้า **x** เป็นคำรหัส เราจะคิดว่าไม่มีข้อผิดพลาดในการส่ง จะสรุปว่า **x** คือคำรหัสที่ส่งมาจากต้นทาง แต่ถ้า **x** ไม่ใช่คำรหัส เราจะรู้ทันทีว่าต้องมีข้อผิดพลาดเกิดขึ้น ในกรณีนี้ เราจะต้องมีหลักเกณฑ์ในการตัดสินว่าคำรหัสใด น่าจะเป็นคำที่ส่งจากต้นทาง โดยปกติรหัส C จะเป็นเซตย่อยของ F_q^n ซึ่งเป็นปริภูมิเวกเตอร์ (ดูรายละเอียดในบทที่ 2) เราจึงมักเรียกสมาชิกใน F_q^n ว่าเวกเตอร์ โดยเฉพาะเมื่อสมาชิกนั้นไม่อยู่ใน C นั่นคือ ไม่เป็นคำรหัส หรือบางครั้งจะเรียกว่า **คำ** (word) เพื่อให้แตกต่างจากคำรหัส

การถอดรหัสโดยใช้ความน่าจะเป็นสูงสุด

(Maximum Likelihood Decoding หรือ MLD)

สมมุติว่าคำรหัสคำหนึ่งจากรหัส C ถูกส่งผ่านช่องสัญญาณ ซึ่งผู้รับปลายทางไม่รู้ว่าเป็นคำรหัสใด เมื่อผู้รับปลายทางได้รับเวกเตอร์ **x** ซึ่งอาจจะเป็นคำรหัสใน C หรือไม่ก็ได้ ผู้รับปลายทางจะคำนวณความน่าจะเป็น

P(รับ x | ส่ง c) สำหรับทุก ๆ คำรหัส c ใน C

และจะสรุปว่า c_0 คือคำรหัสที่ส่งจากต้นทาง ถ้า c_0 คือคำรหัสใน C ที่ทำให้ความน่าจะเป็น P(รับ x | ส่ง c_0) มีค่าสูงสุด นั่นคือ

P(รับ x | ส่ง c_0)

= max{P(รับ x | ส่ง c) | สำหรับทุก ๆ คำรหัส c $\in$ C}

เราเรียกหลักการถอดรหัสแบบนี้ว่า ***การถอดรหัสโดยใช้ความน่าจะเป็นสูงสุด*** การถอดรหัสโดยใช้ความน่าจะเป็นสูงสุด หรือ MLD แบ่งออกเป็น 2 แบบ คือ

1. **การถอดรหัสโดยใช้ความน่าจะเป็นสูงสุดแบบบริบูรณ์**
(Complete Maximum Likelihood Decoding หรือ CMLD)

ถ้า x เป็นคำที่ได้รับ เราหาคำรหัส c_0 ที่ทำให้ความน่าจะเป็น P(รับ x | ส่ง c_0) มีค่าสูงสุด เมื่อเปรียบเทียบกับ P(รับ x | ส่ง c) สำหรับ c ที่เป็นคำรหัสใด ๆ แล้วตัดสินว่า c_0 เป็นคำรหัสที่ส่ง ถอดรหัส x ให้เป็น c_0 สำหรับในกรณีที่มีคำรหัส c มากกว่าหนึ่งคำที่ทำให้ความน่าจะเป็น P(รับ x | ส่ง c) มีค่าสูงสุด ให้เลือกถอดรหัส x ให้เป็นคำใดคำหนึ่งในจำนวนนั้น

2. **การถอดรหัสโดยใช้ความน่าจะเป็นสูงสุดแบบไม่บริบูรณ์**
(Incomplete Maximum Likelihood Decoding หรือ IMLD) เหมือนกับการถอดรหัสแบบ CMLD ยกเว้นในกรณีที่มีคำรหัส c ที่ทำให้ความน่าจะเป็น P(รับ x | ส่ง c) สูงสุดมากกว่าหนึ่งคำ ในกรณีนี้เราจะไม่ถอดรหัส แต่จะขอให้ต้นทางส่งคำรหัสมาใหม่ (retransmit)

ตัวอย่าง 1.8.1 : พิจารณารหัส C_3 = {000000, 101010, 010101, 111111} ในตัวอย่าง 1.7.1 ส่งรหัส C_3 ผ่านช่องสัญญาณ BSC ที่มีความน่าจะเป็นไขว้ p = 0.01 สมมุติว่าปลายทางได้รับ 011101 ซึ่งไม่ใช่คำรหัส เราคำนวณ

ความน่าจะเป็น P(รับ 01110 | ส่ง c) สำหรับทุก ๆ คำรหัส c ใน C_3 ดังนี้

P(รับ 011101 | ส่ง 000000) = $(0.01)^4(0.99)^2$ = $9.8\text{x}10^{-9}$

P(รับ 011101 | ส่ง 010101) = $(0.01)^1(0.99)^5$ =$9.5\text{x}10^{-3}$

P(รับ 011101 | ส่ง 101010) = $(0.01)^5(0.99)^1$ =$9.9\text{x}10^{-11}$

P(รับ 011101 | ส่ง 111111) = $(0.01)^2(0.99)^4$ = $9.6\text{x}10^{-5}$

ถ้าใช้หลักการถอดรหัสแบบ MLD จะเห็นว่า

P(รับ 011101 | ส่ง 010101)

มีค่าสูงสุดเมื่อ c = 010101 เราจึงตัดสินว่า c = 010101 เป็นคำรหัสที่ส่งจากต้นทาง

หมายเหตุ : จากตัวอย่าง 1.8.1 ถ้าต้นทางส่ง 010101 ผ่านช่องสัญญาณที่ทำให้บิตที่สามผิดไป ปลายทางจะได้รับ 011101 การถอดรหัสโดยใช้ MLD จะถอดรหัส 011101 ให้เป็น c = 010101 ซึ่งตรงกับคำที่ส่งจากต้นทาง ชี้ให้เห็นว่าเป็นการถอดรหัสที่ถูกต้อง แต่ถ้าต้นทางส่ง 111111 ผ่านช่องสัญญาณที่ทำให้บิตแรกและบิตที่ห้าผิดไป ปลายทางจะได้รับ 011101 เช่นกัน แต่การถอดรหัสโดยใช้ MLD ในตัวอย่าง 1.8.1 นี้ จะถอดรหัส 011101 ที่ได้รับให้เป็น 010101 เช่นกัน ทั้งนี้เพราะว่า P(รับ 011101 | ส่ง 010101) มีค่าสูงสุด ซึ่งไม่ใช่คำที่ส่ง ชี้ให้เห็นว่าเป็นการถอดรหัสที่ไม่ถูกต้อง แสดงว่าการถอดรหัสไม่จำเป็นจะถูกต้องเสมอไป จะเห็นว่า ถ้าข้อผิดพลาดไม่มากนักการถอดรหัสจะถูกต้อง แต่ถ้าผิดมาก ๆ เป็นไปได้ว่าการถอดรหัสจะไม่ถูกต้อง ถ้าผิดมาก ๆ แสดงว่าช่องสัญญาณที่ใช้ในการสื่อสารเชื่อถือไม่ได้

1.9 ระยะแฮมมิ่ง(Hamming Distance)

ดังได้กล่าวแล้วว่า รหัสส่วนใหญ่ที่เราสนใจ จะเป็นรหัสที่มีฟิลด์จำกัด F_q เป็นชุดตัวอักษร และ F_q^n ก็คือเซตของลำดับที่ยาว n ซึ่ง

แต่ละตำแหน่งในลำดับเป็นสมาชิกของ F_q ดังนั้น $|F_q^n| = q^n$ บางครั้งเราอาจเรียกสมาชิกของ F_q^n ว่า *เวกเตอร์* ทั้งนี้เพราะว่า F_q^n เป็นปริภูมิเวกเตอร์ภายใต้การดำเนินการบวกและการคูณด้วยสเกลาร์ ซึ่งจะได้เห็นรายละเอียดในบทที่ 2 หรือบางครั้งอาจใช้คำว่า คำ แทนเวกเตอร์ เพื่อให้ต่างจากคำว่า **คำรหัส**

นิยาม 1.9.1 ระยะ(แฮมมิ่ง)ระหว่าง x และ y ใน F_q^n ซึ่งแทนด้วย d(x, y) คือจำนวนตำแหน่งใน x และ y ที่แตกต่างกัน

ตัวอย่าง 1.9.1 : ให้ x = 00111และ y = 11001 เป็นสมาชิกใน F_2^5

d(x, y) = d(00111,11001) = 4

ตัวอย่าง 1.9.2 : ให้ x = 0211 และ y = 1220 เป็นสมาชิกใน F_3^4

d(x, y) = d(0211,1220) = 3

หมายเหตุ : ตลอดหนังสือเล่มนี้ เราจะใช้เฉพาะระยะแฮมมิ่งเท่านั้น ดังนั้น เมื่อกล่าวถึงระยะระหว่างเวกเตอร์สองเวกเตอร์ เราจะหมายถึงระยะแฮมมิ่งนี้เท่านั้น

จะเห็นว่า ระยะแฮมมิ่งสอดคล้องกับสมบัติต่อไปนี้

1. d(x, y) = 0 ก็ต่อเมื่อ x = y
2. d(x, y) = d(y, x) สำหรับ x, y ใด ๆ ใน F_q^n
3. $d(x, y) \leq d(x, z) + d(z, y)$ สำหรับ x, y, z ใด ๆ ใน F_q^n

นั่นคือ ระยะแฮมมิ่งเป็น ***เมตริก***(metric) บนเซต F_q^n เราเรียกสมบัติข้อที่ 3 นี้ว่า ***อสมการสามเหลี่ยม*** (triangle inequality)

ตัวอย่าง 1.9.3 : ให้ x = 0011 เป็นเวกเตอร์ใน F_2^4 จงหา y ใน F_2^4 ซึ่ง d(x, y) = 1

วิธีทำ ในที่นี้เราต้องการหาเวกเตอร์ใน F_2^4 ที่ต่างจาก x = 0011 หนึ่งบิต บิตใดก็ได้ จะเห็นว่า

1011 ต่างจาก **x** = 0011 ในบิตแรก ดังนั้น d(1011, 0011) = 1

0111 ต่างจาก **x** = 0011 ในบิตที่สอง ดังนั้น d(0111, 0011) = 1

0001 ต่างจาก **x** = 0011 ในบิตที่สาม ดังนั้น d(0001, 0011) = 1

0010 ต่างจาก **x** = 0011 ในบิตที่สี่ ดังนั้น d(0010, 0011) = 1

ดังนั้น จำนวนเวกเตอร์ **y** ซึ่ง d(**x**, **y**) = 1 มีทั้งหมด 4 เวกเตอร์คือ 1011, 0111, 0001 และ 0010

ตัวอย่าง 1.9.4 : ให้ **x** = 0011 เป็นเวกเตอร์ใน F_2^4 จงหา **y** ใน F_2^4 ซึ่ง d(**x**, **y**) = 2

1111 ต่างจาก **x** = 0011 ในบิตที่ 1 และ 2 ดังนั้น d(1111, 0011) = 2

1001 ต่างจาก **x** = 0011 ในบิตที่ 1 และ 3 ดังนั้น d(1001, 0011) = 2

1010 ต่างจาก **x** = 0011 ในบิตที่ 1 และ 4 ดังนั้น d(1011, 0011) = 2

0101 ต่างจาก **x** = 0011 ในบิตที่ 2 และ 3 ดังนั้น d(0101, 0011) = 2

0110 ต่างจาก **x** = 0011 ในบิตที่ 2 และ 4 ดังนั้น d(0110, 0011) = 2

0000 ต่างจาก **x** = 0011 ในบิตที่ 3 และ 4 ดังนั้น d(0000, 0011) = 2

ดังนั้น จำนวนเวกเตอร์ **y** ซึ่ง d(**x**, **y**) = 2 มีทั้งหมด 6 เวกเตอร์ คือ

1111, 1001, 1010, 0101, 0110 และ 0000

จากตัวอย่าง 1.9.3 และ 1.9.4 จะเห็นว่า

จำนวนเวกเตอร์ที่ต่างจาก **x** หนึ่งบิต = จำนวนวิธีเลือก 1 บิต จาก 4 บิต

$$= \binom{4}{1} = 4$$

จำนวนเวกเตอร์ที่ต่างจาก **x** สองบิต = จำนวนวิธีเลือก 2 บิต จาก 4 บิต

$$= \binom{4}{2} = 6$$

ในกรณีทั่วไป ถ้า $\mathbf{x} \in F_2^n$ แล้วจำนวนเวกเตอร์ **y** ใน F_2^n ที่ทำให้ d(**x**, **y**) = i เท่ากับจำนวนวิธีเลือก i บิต จาก n บิต คือเท่ากับ $\binom{n}{i}$

ตัวอย่าง 1.9.5 : ให้ $\mathbf{x}$ = 0211 เป็นเวกเตอร์ใน F_3^4 จงหา $\mathbf{y}$ ใน F_3^4 ซึ่ง $d(\mathbf{x}, \mathbf{y}) = 1$

วิธีทำ ในที่นี้เราต้องการหาเวกเตอร์ใน F_3^4 ที่ต่างจาก $\mathbf{x}$ = 0211 หนึ่งตำแหน่ง ตำแหน่งใดก็ได้ จะเห็นว่า

1211 และ 2211 ต่างจาก $\mathbf{x}$ = 0211 ในตำแหน่งแรก

0011 และ 0111 ต่างจาก $\mathbf{x}$ = 0211 ในตำแหน่งที่สอง

0201 และ 0221 ต่างจาก $\mathbf{x}$ = 0211 ในตำแหน่งที่สาม

0210 และ 0212 ต่างจาก $\mathbf{x}$ = 0211 ในตำแหน่งที่สี่

ดังนั้น จำนวนเวกเตอร์ $\mathbf{y}$ ซึ่ง $d(\mathbf{x}, \mathbf{y}) = 1$ มีทั้งหมด 4 x 2 = 8 เวกเตอร์ ได้แก่เวกเตอร์

1211, 2211, 0011, 0111, 0201, 0221, 0210 และ 0212

ในตัวอย่าง 1.9.5 จะเห็นว่า เมื่อพิจารณาตำแหน่งที่ i ใด ๆ ของเวกเตอร์ $\mathbf{x} \in F_3^4$ จำนวนเวกเตอร์ที่ต่างจาก $\mathbf{x}$ ในตำแหน่งที่ i ใด ๆ เท่ากับ 3 – 1 = 2 เวกเตอร์ และเนื่องจากเวกเตอร์แต่ละเวกเตอร์มี 4 ตำแหน่ง จึงเลือกตำแหน่งที่แตกต่างหนึ่งตำแหน่งได้

$$\binom{4}{1} = 4$$

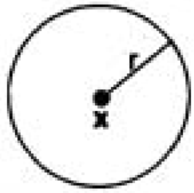
วิธี ดังนั้น จำนวนเวกเตอร์ที่แตกต่างจาก $\mathbf{x}$ หนึ่งตำแหน่ง มีทั้งหมด

$$\binom{4}{1} \times (3-1) = 4 \times 2 = 8 \text{ เวกเตอร์}$$

ในกรณีทั่วไป จำนวนเวกเตอร์ที่ต่างจาก $\mathbf{x} \in F_q^n$ ในแต่ละตำแหน่ง มี q – 1 เวกเตอร์ และเลือกตำแหน่งที่แตกต่าง i ตำแหน่งจากทั้งหมด n ตำแหน่ง ได้ $\binom{n}{i}$ วิธี ดังนั้น จำนวนเวกเตอร์ที่แตกต่างจาก $\mathbf{x} \in F_q^n$ เป็นจำนวน i ตำแหน่งใด ๆ เท่ากับ

$$\binom{n}{i}(q-1)^i$$

เวกเตอร์



รูป 1.9.1 ทรงกลม S(x, r)

นิยาม 1.9.2

> สำหรับ $\mathbf{x} \in F_q^n$ และ r ที่เป็นจำนวนเต็มที่ไม่เป็นลบ ***ทรงกลม*** (sphere) ที่มีจุดศูนย์กลางที่ $\mathbf{x}$ และมีรัศมี r คือเซต
>
> $$S(\mathbf{x}, r) = \{ \mathbf{y} \in F_q^n \mid d(\mathbf{x}, \mathbf{y}) \leq r \}$$

เราสามารถนับจำนวนเวกเตอร์ใน F_q^n ซึ่งบรรจุอยู่ในทรงกลม $S(\mathbf{x}, r)$ (ในรูป 1.9.1) ได้ ดังในทฤษฎีบทประกอบต่อไปนี้

ทฤษฎีบทประกอบ 1.9.1

> จำนวนเวกเตอร์ที่บรรจุในทรงกลม $S(\mathbf{x}, r)$ ที่มีจุดศูนย์กลางที่ $\mathbf{x} \in F_q^n$ และมีรัศมี r คือ
>
> $$\binom{n}{0}+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\cdots+\binom{n}{r}(q-1)^r$$
>
> ในกรณีที่ $q = 2$ จำนวนเวกเตอร์ที่บรรจุในทรงกลม $S(\mathbf{x}, r)$ คือ
>
> $$\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\cdots+\binom{n}{r}$$

พิสูจน์ เวกเตอร์ที่บรรจุอยู่ในทรงกลม $S(\mathbf{x}, r)$ ประกอบด้วยเวกเตอร์ $\mathbf{x}$ ที่เป็นจุดศูนย์กลางเอง ซึ่งมี $\binom{n}{0} = 1$ เวกเตอร์ และ

เวกเตอร์ที่อยู่ห่างจาก x เป็นระยะเท่ากับ 1 มี $\binom{n}{1}(q-1)$ เวกเตอร์

เวกเตอร์ที่อยู่ห่างจาก x เป็นระยะเท่ากับ 2 มี $\binom{n}{2}(q-1)^2$ เวกเตอร์

$$\vdots$$

เวกเตอร์ที่อยู่ห่างจาก x เป็นระยะเท่ากับ r มี $\binom{n}{r}(q-1)^r$ เวกเตอร์

ดังนั้น จำนวนเวกเตอร์ที่ห่างจาก x เป็นระยะน้อยกว่าหรือเท่ากับ r มีทั้งหมด

$$\binom{n}{0}+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\cdots+\binom{n}{r}(q-1)^r$$

เวกเตอร์ และถ้า q = 2 เราได้ จำนวนเวกเตอร์ที่ห่างจาก x เป็นระยะน้อยกว่าหรือเท่ากับ r มีทั้งหมด

$$\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\cdots+\binom{n}{r}$$

ตามที่ต้องการจะแสดง ▬

ตัวอย่าง 1.9.6 : ใน F_3^4 จำนวนเวกเตอร์ที่บรรจุในทรงกลม S(1022, 2) คือ

$$\binom{4}{0}+\binom{4}{1}(3-1)+\binom{4}{2}(3-1)^2 = 1 + 4\text{x}2 + 6\text{x}2^2 = 33$$

ตัวอย่าง 1.9.7 : ใน F_2^5 จำนวนเวกเตอร์ที่บรรจุในทรงกลม S(10010, 2) คือ

$$\binom{5}{0}+\binom{5}{1}+\binom{5}{2} = 1 + 5 + 10 = 16$$

ข้อสังเกต : จำนวนเวกเตอร์ที่บรรจุในทรงกลมซึ่งบางครั้งจะเรียกว่าขนาดของทรงกลม จะขึ้นอยู่กับขนาดของรัศมี r เท่านั้น ไม่ขึ้นอยู่กับจุดศูนย์กลาง นั่นคือ

ทรงกลมใดที่มีรัศมีเท่ากัน คำที่บรรจุในทรงกลมนั้น ๆ จะมีจำนวนเท่ากัน ดังนั้นเราจะแทนขนาดทรงกลมใน F_q^n ที่มีรัศมีเท่ากับ r ด้วย $V_q^n(r)$

นิยาม 1.9.3 ระยะน้อยสุด (minimum distance) ของรหัส C ซึ่งจะเขียนแทนด้วย d(C) คือ

$$d(C) = \min\{d(x, y) \mid x, y \in C \text{ และ } x \neq y\}$$

ตัวอย่าง 1.9.8 : จากตัวอย่าง 1.4.1, 1.4.2 และ 1.4.3 เรามี

$C_1 = \{000,111\}$

$C_2 = \{000, 011, 101, 110\}$

$C_3 = \{000000, 010101, 101010, 111111\}$

เห็นได้ชัดว่า $d(C_1) = 3$ หาระยะระหว่างคำรหัสใน C_2 แต่ละคู่ เราได้

ตาราง 1.9.1 : ระยะระหว่างคำรหัสแต่ละคู่ใน C_2

คำรหัส x	คำรหัส y	d(x,y)
000	011	2
000	101	2
000	110	2
011	101	2
011	110	2
101	110	2

ผลดังในตาราง 1.9.1 จะพบว่าระยะระหว่างเวกเตอร์แต่ละคู่เท่ากันทั้ง

ตาราง 1.9.2 : ระยะระหว่างคำรหัสแต่ละคู่ใน C_3

คำรหัส x	คำรหัส y	d(x,y)
000000	010101	3
000000	101010	3
000000	111111	6
010101	101010	6
010101	111111	3
101010	111111	3

หมด คือเท่ากับ 2 ดังนั้น ระยะที่น้อยที่สุดเท่ากับ 2 นั่นคือ $d(C_2) = 2$ ในทำนองเดียวกัน เราสามารถตรวจสอบได้ไม่ยากจากตาราง 1.9.2 ว่า $d(C_3) = 3$

ตัวอย่าง 1.9.9 : ให้ C_4 = {012210, 112112, 221020} หาระยะระหว่างคำรหัสใน C_4 แต่ละคู่ เราได้ผลดังในตาราง 1.9.3

ตาราง 1.9.3 : ระยะระหว่างคำรหัสแต่ละคู่ใน C_4

คำรหัส x	คำรหัส y	d(x,y)
012210	112112	3
012210	221020	5
112112	221020	6

ดังนั้น $d(C_4) = 3$

นิยาม 1.9.4

> เราจะเรียกรหัสที่มีความยาว n มีขนาด M และมีระยะน้อยสุด d ว่า รหัส-(n, M, d) และเรียก n, M, d ว่าตัวแปรของรหัส

ตัวอย่าง 1.9.10 : รหัส C_1 ในตัวอย่าง 1.9.8 เป็นรหัสไบนารี-(3,2,3)
รหัส C_2 ในตัวอย่าง 1.9.8 เป็นรหัสไบนารี-(3,4,2)
รหัส C_3 ในตัวอย่าง 1.9.8 เป็นรหัสไบนารี-(6,4,3)
รหัส C_4 ในตัวอย่าง 1.9.9 เป็นรหัสเทอร์นารี-(6,3,3)

เราจะได้เห็นต่อไปว่าระยะน้อยสุดของรหัส C ใด ๆ หรือ d(C) เป็นเครื่องมือที่สำคัญ ในการกำหนดความสามารถในการตรวจจับหรือการแก้ไขข้อผิดพลาดของรหัสนั้น

ทฤษฎีบทประกอบ 1.9.2

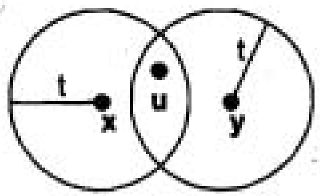
> ถ้า $d(C) \geq 2t + 1$ และสำหรับ $x, y \in C$ ซึ่ง $x \neq y$ แล้ว $S(x, t) \cap S(y, t)$ จะเป็นเซตว่าง

พิสูจน์ พิจารณาทรงกลม $S(x, t)$ และ $S(y, t)$ เมื่อ $x, y \in C$ และ $x \neq y$ ถ้า $S(x, t)$ และ $S(y, t)$ มีสมาชิกร่วมกัน สมมุติให้

$$u \in S(x, t) \cap S(y, t)$$

นั่นคือ $u \in S(x, t)$ และ $u \in S(y, t)$ แสดงว่า

$$d(x, u) \leq t \text{ และ } d(u, y) \leq t$$



ใช้อสมการสามเหลี่ยม เราได้

$$d(x, y) \leq d(x, u) + d(u, y) \leq t + t = 2t$$

ซึ่งเป็นไปไม่ได้ เพราะ $d(C) \geq 2t + 1$ ดังนั้น เราสรุปได้ว่า $S(x, t)$ และ $S(y, t)$ ไม่มีสมาชิกร่วมกัน ■

ทฤษฎีบท 1.9.1

Sphere-packing bound หรือ Hamming bound

ถ้า $C \subset F_q^n$ เป็นรหัส-$(n, M, 2t + 1)$ แล้ว

$$M\left\{\binom{n}{0}+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\ldots+\binom{n}{t}(q-1)^t\right\} \leq q^n$$

และสำหรับ $q = 2$

$$M\left\{\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\ldots+\binom{n}{t}\right\} \leq 2^n$$

พิสูจน์ เป็นผลโดยตรงจาก $|F_q^n| = q^n$, $|C| = M$ และทฤษฎีบทประกอบ1.9.1 และ 1.9.2 ■

หมายเหตุ : ถ้าให้ $V_q^n(t)$ แทน

$$\binom{n}{0}+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\ldots+\binom{n}{t}(q-1)^t$$

เราสามารถเขียนอสมการในทฤษฎีบท 1.9.1 ใหม่ได้

$$M \le \frac{q^n}{\sum_{i=0}^{t}\binom{n}{i}(q-1)^i} = \frac{q^n}{V_q^n(t)}$$

และ

$$M \le \frac{2^n}{\sum_{i=0}^{t}\binom{n}{i}} = \frac{2^n}{V_2^n(t)}$$

ตัวอย่าง 1.9.11 : ให้ C เป็นรหัสฐานสองที่มีความยาวเท่ากับ 6 และมีระยะน้อยสุด d = 3 ต้องการจะหาขอบเขตบนของจำนวนคำรหัสใน C

วิธีทำ ในที่นี้ n = 6, d = 3 = 2t + 1 เราได้ t = 1 จากทฤษฎีบท 1.9.1 เราได้

$$M \le \frac{2^6}{\binom{6}{0}+\binom{6}{1}} = \frac{64}{7} \approx 9.12$$

แสดงว่าคำรหัสใน C จะมีได้ไม่เกิน 9 คำ

นิยาม 1.9.5

> น้ำหนักแฮมมิ่ง(Hamming weight) ของ $x \in F_q^n$ หรือเรียกสั้น ๆ ว่า น้ำหนักของ x คือจำนวนตำแหน่งที่ไม่เป็น 0 ใน x ซึ่งจะเขียนแทนด้วย wt(x)

ตัวอย่าง 1.9.12 :

1. ถ้า x = 010101 $\in F_2^6$ แล้ว wt(x) = wt(010101) = 3
2. ถ้า x = 10220 $\in F_3^5$ แล้ว wt(x) = wt(10220) = 3

ข้อสังเกต : ถ้า **x** ที่เป็นเวกเตอร์ใด ๆ ใน F_2^n แล้ว wt(**x**) = d(**x**, **0**) เมื่อ **0** คือเวกเตอร์ศูนย์ใน F_2^n

---

ตัวอย่าง 1.9.13 :

1. **x** = 010101, wt(**x**) = 3 = d(010101, 000000)
2. **y** = 10220, wt(**y**) = 3 = d(10220, 00000)

นิยาม 1.9.6 น้ำหนักน้อยสุด (minimum weight) ของรหัส C ซึ่งเขียนแทน ด้วย wt(C) คือ

$$wt(C) = \min\{wt(\mathbf{x}) \mid \mathbf{x} \in C \text{ และ } \mathbf{x} \neq \mathbf{0}\}$$

หมายเหตุ : wt(C) เป็นน้ำหนักที่น้อยที่สุด ในระหว่างคำรหัสใน C ที่ไม่ใช่ศูนย์

ตัวอย่าง 1.9.14 : ให้ C_1, C_2, C_3 เป็นรหัสในตัวอย่าง 1.9.8 และ C_4 เป็นรหัสในตัวอย่าง 1.9.9 เราได้

$$wt(C_1) = 3,\ wt(C_2) = 2,\ wt(C_3) = 3 \text{ และ } wt(C_4) = 4$$

ตาราง 1.9.3

x	0	1
0	0	0
1	0	1

นิยาม 1.9.7 ให้ $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ และ $\mathbf{y} = (y_1, y_2, \ldots, y_n) \in F_2^n$ ส่วนร่วมของ **x** และ **y** ซึ่งเขียนแทนด้วย $\mathbf{x} \cap \mathbf{y}$ คือ

$$\mathbf{x} \cap \mathbf{y} = (x_1y_1, x_2y_2, \ldots, x_ny_n)$$

เมื่อ x_iy_i คือผลคูณซึ่งกำหนดในตาราง1.9.3 สำหรับ i = 1, 2, …, n

---

ตัวอย่าง 1.9.15 : ให้ **x** = 100111 และ **y** = 111001 เราได้

$$\mathbf{x} \cap \mathbf{y} = 100001$$

ข้อสังเกต : ถ้า $x = (x_1, x_2, \ldots, x_n)$ และ $y = (y_1, y_2, \ldots, y_n)$ เป็นเวกเตอร์ใน F_2^n แล้ว ตำแหน่งใน $x \cap y$ ที่เป็น 1 คือตำแหน่งของ x และ y ที่เป็น 1 ตรงกัน ส่วนตำแหน่งอื่น ๆ จะเป็น 0 ทั้งหมด

ทฤษฎีบท 1.9.2

ถ้า $x, y \in F_2^n$ แล้ว $d(x, y) = wt(x) + wt(y) - 2wt(x \cap y)$

พิสูจน์ เนื่องจาก

d(x, y) = (จำนวนเลข1 ใน x) + (จำนวนเลข1 ใน y)

− 2(จำนวนตำแหน่งใน x และ y ที่มี 1 ตรงกัน)

ดังนั้น $d(x, y) = wt(x) + wt(y) - 2wt(x \cap y)$ ∎

1.10 การถอดรหัสให้เป็นคำรหัสที่ใกล้ที่สุด (Nearest Neighbour Decoding)

สมมุติว่าคำรหัสใน C ถูกส่งผ่านช่องสัญญาณที่มีสิ่งรบกวน และสมมุติว่า x เป็นเวกเตอร์ที่ได้รับปลายทาง ผู้รับปลายทางหรือเครื่องถอดรหัส จะถอดรหัส x ให้เป็นคำรหัส c เมื่อ c คือคำรหัสใน C ที่อยู่ใกล้ x ที่สุด เรียกวิธีถอดรหัสโดยใช้หลักเกณฑ์ดังกล่าวนี้ว่า ***การถอดรหัสให้เป็นคำรหัสที่ใกล้ที่สุด*** ดังนั้น เมื่อรับเวกเตอร์ x เราจะเปรียบเทียบเวกเตอร์ที่ได้รับกับคำรหัสทั้งหมดที่มีอยู่ เพื่อหาคำรหัสที่อยู่ใกล้ x ที่สุด

ตัวอย่าง 1.10.1 : พิจารณารหัส C_3 = {000000, 010101, 101010, 111111} ในตัวอย่าง 1.9.8 สมมุติว่าปลายทางได้รับเวกเตอร์ x = 011101 เราหา d(x, c) สำหรับทุก ๆ c ใน C ดูตาราง 1.10.1 จะพบว่า

$$d(x, 010101) = 1$$

ซึ่งเป็นระยะที่น้อยที่สุด แสดงว่า x อยู่ใกล้คำรหัส 010101 ที่สุด

ตาราง 1.10.1

คำที่รับ x	คำรหัส c	d(x,c)
011101	000000	4
	101010	5
	010101	1
	111111	2

ดังนั้น เราถอดรหัส **x** ให้เป็นคำรหัส **c** = 010101

ให้ **x** และ **y** เป็นคำที่มีความยาว n ในหัวข้อ 1.5 เรารู้ว่า ถ้า **x** = **y** แล้ว

$$P(\text{รับ } \mathbf{y} \mid \text{ส่ง } \mathbf{x}) = (1-p)^n$$

และถ้า **x** และ **y** แตกต่างกัน i ตำแหน่ง จะได้

$$P(\text{รับ } \mathbf{y} \mid \text{ส่ง } \mathbf{x}) = p^i(1-p)^{n-i}$$

ดังนั้น ถ้า $p < \frac{1}{2}$ เราได้ทฤษฎีบทต่อไปนี้

ทฤษฎีบท 1.10.1

> สำหรับช่องสัญญาณ BSC ที่มี $p < \frac{1}{2}$ วิธีการถอดรหัสโดยใช้ความน่าจะเป็นสูงสุด และวิธีการถอดรหัสให้เป็นคำรหัสที่ใกล้ที่สุดเป็นวิธีที่ให้ผลเหมือนกัน

พิสูจน์ ให้ C เป็นรหัสที่ใช้ในช่องสัญญาณ BSC และสมมุติให้ n เป็นความยาวของรหัส C เราทราบว่า

$$d(\mathbf{x}, \mathbf{c}) = i \text{ ก็ต่อเมื่อ } p(\text{รับ } \mathbf{x} \mid \text{ส่ง } \mathbf{c}) = p^i(1-p)^{n-i}$$

และเนื่องจาก $p < \frac{1}{2}$ เราพบว่า

$$(1-p)^n > p(1-p)^{n-1} > p^2(1-p)^{n-2} > \ldots > p^{n-1}(1-p) > p^n$$

กล่าวคือ

ความน่าจะเป็นที่ผิด 0 บิต > ความน่าจะเป็นที่ผิด 1 บิต

> ความน่าจะเป็นที่ผิด 2 บิต > ... > ความน่าจะเป็นที่ผิด n บิต

นั่นคือ ความน่าจำเป็นที่จะผิดน้อยตำแหน่ง มีค่ามากกว่าความน่าจะเป็นที่จะผิดมากตำแหน่ง ดังนั้น จึงสรุปได้ว่า วิธีการถอดรหัสโดยใช้ความน่าจะเป็นสูงสุด และวิธีการถอดรหัสให้เป็นคำรหัสที่ใกล้ที่สุดเป็นวิธีที่ให้ผลเหมือนกัน ■

จะเห็นว่าการหาระยะระหว่างสองเวกเตอร์ ง่ายกว่าการหาความน่าจะเป็น ดังนั้น เราจึงนิยมใช้การถอดรหัสโดยใช้คำรหัสที่ใกล้ที่สุดมากกว่าจะใช้วิธีการถอดรหัสโดยใช้ความน่าจะเป็นสูงสุด การถอดรหัสให้เป็นคำรหัสที่ใกล้ที่สุด แบ่งออกเป็นสองแบบเช่นเดียวกับวิธีการถอดรหัสโดยใช้ความน่าจะเป็นสูงสุด คือแบ่งเป็นการถอดรหัสแบบบริบูรณ์และแบบไม่บริบูรณ์

ตัวอย่าง 1.10.2 : พิจารณารหัส C_3 = {000000, 010101, 101010, 111111} ในตัวอย่าง 1.10.1 ถ้า **x** = 011101 เป็นเวกเตอร์ที่ได้รับ จะเห็นว่าคำรหัส 010101 เป็นคำเดียวที่อยู่ใกล้ **x** มากที่สุด ดังนั้น เครื่องถอดรหัสที่ใช้หลักเกณฑ์การถอดรหัสให้เป็นคำรหัสที่ใกล้ที่สุด จะถอดรหัส **x** ให้เป็นคำรหัส 010101

ตัวอย่าง 1.10.3 : พิจารณารหัส C = {0000, 1010, 0111}
สมมุติว่า **x** = 1000 เป็นเวกเตอร์ที่ได้รับ ระยะระหว่าง **x** และคำรหัสใน

ตาราง 1.10.2

คำที่รับ **x**	คำรหัส **c**	d(x, c)
1000	0000	1
	1010	1
	0111	4

C ปรากฏดังในตาราง 1.10.2 จะเห็นว่า มีคำรหัสที่อยู่ใกล้ x ที่สุด 2 คำ คือ 0000 และ 1010 ถ้าใช้การถอดรหัสแบบบริบูรณ์ เราจะเลือกคำใดคำหนึ่ง แล้วถอดรหัส x ให้เป็นคำนั้น เช่นเลือกถอดรหัส x ให้เป็น 0000 แต่ถ้าใช้การถอดรหัสแบบไม่บริบูรณ์ เราจะไม่ตัดสินว่าคำใดเป็นคำที่ส่งจากต้นทาง แต่จะขอให้ต้นทางส่งข้อมูลมาใหม่

ตัวอย่าง 1.10.4 : พิจารณารหัส $C_1 = \{000, 111\}$ จงหาความน่าจะเป็นที่จะถอดรหัสผิดพลาด

วิธีทำ สมมุติว่า 000 เป็นคำรหัสที่ส่ง เราจะถอดรหัสได้ถูกต้องถ้าเราได้รับเวกเตอร์ 000, 100, 010 หรือ 001 ดังนั้น ความน่าจะเป็นที่จะถอดรหัสคำที่ได้รับได้ถูกต้อง ตรงกับคำรหัสที่ส่งจากต้นทาง คือ

$$(1-p)^3+3p(1-p)^2 = (1-p)^2(1+2p)$$

จะเห็นว่า ถ้า 111 เป็นคำที่ส่ง ความน่าจะเป็นที่จะถอดรหัสคำที่ได้รับได้ถูกต้องจะมีค่าเท่ากัน ดังนั้น ความน่าจะเป็นที่จะถอดคำรหัสผิดพลาด (word error probability) คือ

$$P_{err}(C) = 1-(1-p)^2(1+2p) = 3p^2-2p^3$$

1.11 รหัสตรวจจับและแก้ไขข้อผิดพลาด(error detecting and correcting code)

เมื่อได้รับเวกเตอร์ **y** เราจะเปรียบเทียบ **y** กับคำรหัสทั้งหมด ถ้า **y** ไม่ตรงกับคำรหัสใดเลย เราบอกได้ทันทีว่าต้องมีข้อผิดพลาดเกิดขึ้น เพราะคำที่ส่งจากต้นทางต้องเป็นคำรหัสเท่านั้น เราเรียกรหัสที่มีความสามารถดังกล่าวนี้ว่า ***รหัสตรวจจับข้อผิดพลาด***

ถ้าเราใช้ $C_0 = \{00, 01, 10, 11\}$ เป็นรหัส ไม่ว่าต้นทางจะส่งคำรหัสใดใน C_0 ถ้ามีข้อผิดพลาดเกิดขึ้นแม้เพียงตำแหน่งเดียว คำที่ปลายทางได้รับจะเป็นคำรหัส ทำให้ผู้รับปลายทางเข้าใจผิดว่าคำรหัสที่รับมานั้นเป็นคำที่ส่งจากต้นทาง แสดงว่ารหัส C_0 ไม่สามารถตรวจ

จับข้อผิดพลาดได้เลย เหตุที่เป็นเช่นนี้เพราะคำรหัสแต่ละคำใน C_0 มีระยะใกล้กันเกินไป

พิจารณารหัส C_2 = {000, 011, 101, 110} สมมุติว่าต้นทางส่งคำรหัส 011 และสมมุติว่าบิตที่สองผิดไปเป็น 0 ผู้รับปลายทางจะได้รับ 001 ซึ่งไม่ตรงกับคำรหัสคำใดเลยใน C_2 ผู้รับปลายทางจะรู้ว่าต้องมีข้อผิดพลาดเกิดขึ้น แต่ไม่รู้ว่าผิดที่ตำแหน่งใด จริง ๆ แล้วเราสามารถตรวจสอบได้ว่า ไม่ว่าต้นทางจะส่งคำรหัสใดใน C_2 ไม่จำเป็นต้องเป็น 011 ถ้าคำที่รับผิดไปจากคำที่ส่ง ในตำแหน่งใดตำแหน่งหนึ่งเพียงตำแหน่งเดียว คำที่ปลายทางได้รับ จะไม่ตรงกับคำรหัสใดเลย ดังนั้นผู้รับรู้ว่าต้องมีข้อผิดพลาดเกิดขึ้น แสดงว่ารหัส C_2 มีความสามารถในการตรวจจับข้อผิดพลาดได้หนึ่งตำแหน่ง

สมมุติว่าต้นทางส่ง 011 ใน C_2 เช่นเดิม แต่สมมุติว่ามีข้อผิดพลาดเกิดขึ้นสองตำแหน่ง สมมุติว่าผิดพลาดในตำแหน่งแรกและตำแหน่งที่สอง ผู้รับปลายทางจะได้รับ 101 ซึ่งตรงกับคำรหัสอีกคำหนึ่งใน C_2 ผู้รับจะเข้าใจผิดว่า 101 เป็นคำที่ส่ง ซึ่งเป็นการถอดรหัสที่ผิดพลาด แสดงว่าถ้ามีข้อผิดพลาดเกิดขึ้นสองตำแหน่ง รหัส C_2 จะไม่สามารถตรวจจับข้อผิดพลาดได้ ในกรณีนี้ เรากล่าวว่า รหัส C_2 สามารถตรวจจับข้อผิดพลาดได้ถึง 1 ตำแหน่ง เพราะถ้าผิดมากกว่า 1 ตำแหน่งแล้ว C_2 จะไม่สามารถตรวจจับข้อผิดพลาดได้

นิยาม 1.11.1 จะกล่าวว่ารหัส C สามารถตรวจจับข้อผิดพลาดได้ถึง t ตำแหน่ง ถ้ามีข้อผิดพลาดน้อยกว่าหรือเท่ากับ t ตำแหน่ง คำที่รับจะไม่ใช่คำรหัส

ตัวอย่าง 1.11.1 : เห็นได้ชัดว่ารหัส C_1 = {000, 111} สามารถตรวจจับข้อผิดพลาดได้ถึง 2 ตำแหน่ง ทั้งนี้เพราะว่า ไม่ว่าต้นทางจะส่งคำรหัส 000 หรือ 111 ถ้ามีข้อผิดพลาด 1 หรือ 2 ตำแหน่ง คำที่ได้รับจะไม่ใช่คำรหัส ทำให้ผู้

รับปลายทางรู้ว่ามีข้อผิดพลาดเกิดขึ้น แต่ถ้ามีข้อผิดพลาด 3 ตำแหน่ง C_1 จะไม่สามารถตรวจจับข้อผิดพลาดได้

ข้อสังเกต : $d(C_0) = 1$, $d(C_1) = 3$, $d(C_2) = 2$ และ $d(C_3) = 3$ เราพบว่า C_0 ไม่สามารถตรวจจับข้อผิดพลาดได้ C_2 ตรวจจับข้อผิดพลาดได้ 1 ตำแหน่ง ส่วน C_1 และ C_3 สามารถตรวจจับข้อผิดพลาดได้ถึง 2 ตำแหน่ง

ทฤษฎีบท 1.11.1 รหัส C สามารถตรวจจับข้อผิดพลาดได้ถึง t ตำแหน่ง ก็ต่อเมื่อ $d(C) \geq t + 1$

พิสูจน์ สมมุติให้รหัส C สามารถตรวจจับข้อผิดพลาดได้ถึง t ตำแหน่ง สมมุติว่า **y** เป็นคำที่ได้รับซึ่งผิดจากคำรหัสที่ส่ง t ตำแหน่งหรือน้อยกว่า แสดงว่า **y** ต้องไม่ใช่คำรหัส นั่นคือ คำรหัสแต่ละคู่ต้องต่างกันมากกว่า t ตำแหน่ง ดังนั้น $d(C) \geq t + 1$

ในทางกลับกัน สมมุติให้ $d(C) \geq t + 1$ จะแสดงว่ารหัส C สามารถตรวจจับข้อผิดพลาดได้ถึง t ตำแหน่ง ให้ **x** เป็นคำรหัส ที่ส่ง สมมุติว่า **y** เป็นคำที่ได้รับซึ่งผิดไปจากคำที่ส่ง t ตำแหน่งหรือน้อยกว่า ดังนั้น $d(\mathbf{x}, \mathbf{y}) \leq t$ แสดงว่า **y** ต้องไม่ใช่คำรหัส เพราะว่า $d(C) \geq t + 1$ นั่นคือรหัส C สามารถตรวจจับข้อผิดพลาดได้ถึง t ตำแหน่ง ∎

ตัวอย่าง 1.11.2 : พิจารณารหัส C_3 = {000000, 010101, 101010, 111111}

สมมุติว่า 010101 เป็นคำรหัสที่ส่ง และสมมุติว่ามีข้อผิดพลาดเกิดขึ้นในตำแหน่งที่ห้า ดังนั้น ผู้รับจะได้รับ 010111 ซึ่งเมื่อเปรียบเทียบกับคำรหัสแต่ละคำใน C_3 จะเห็นว่าคำที่ได้รับไม่ตรงกับคำรหัสคำใดเลย แสดงว่ามีข้อผิดพลาดเกิดขึ้น นั่นคือ รหัส C_3 สามารถตรวจจับข้อผิดพลาดได้ จากทฤษฎีบท 1.11.1 และเนื่องจาก $d(C_3) = 3$ (ดู

ตัวอย่าง 1.9.8) แสดงว่า C_3 สามารถตรวจจับข้อผิดพลาดได้ถึง 2 ตำแหน่ง

รหัส C_3 นอกจากจะตรวจจับข้อผิดพลาดได้ถึง 2 ตำแหน่งแล้ว ยังสามารถแก้ไขข้อผิดพลาดได้อีกด้วย สมมุติว่า 010101 เป็นคำรหัสที่ส่ง ถ้ามีข้อผิดพลาดเกิดขึ้นหนึ่งตำแหน่ง ตำแหน่งใดก็ได้ คำที่ปลายทางได้รับคือ

110101, 000101, 011101, 010001, 010111 หรือ 010100

ตาราง 1.11.1

คำที่รับ	คำรหัส			
	000000	010101	101010	111111
110101	4	1	5	2
000101	2	1	5	4
011101	4	1	5	2
010001	2	1	5	4
010111	4	1	5	2
010100	2	1	5	4

ไม่ว่าจะได้รับคำใด คำที่ได้รับนั้นจะอยู่ใกล้คำรหัส 010101 ที่สุด(ดูตาราง 1.11.1) ถ้าใช้หลักการถอดรหัสให้เป็นคำรหัสที่ใกล้ที่สุด เราจะถอดรหัสคำที่ได้รับให้เป็น 010101 ซึ่งตรงกับคำที่ส่ง

ผู้อ่านสามารถตรวจสอบได้เองว่า ไม่ว่าคำรหัสที่ส่งจะเป็นคำใดใน C_3 และถ้ามีข้อผิดพลาดเกิดขึ้น 1 ตำแหน่ง ผู้รับปลายทางสามารถแก้ไขให้ถูกต้องได้ ในกรณีนี้ชี้ให้เห็นว่า เมื่อมีข้อผิดพลาด 1 ตำแหน่ง ตำแหน่งใดก็ได้ ผู้รับปลายทางหรือเครื่องถอดรหัสสามารถแก้ไขให้ถูกต้องได้

ในกรณีนี้ ถ้าเราหาสมาชิกในทรงกลม S(c,1) สำหรับแต่ละ c ใน C_3 จะพบว่า

S(000000,1) = {000000,100000, 010000, 001000, 000100, 000010, 000001}

S(010101,1) = {010101,110101, 000101, 011101, 010001, 010111, 010100}

S(101010,1) = {101010, 001010, 111010, 100010, 101110, 101000,101011}

S(111111,1) = {111111, 011111, 101111, 110111, 111011, 111101,111110}

จะเห็นว่า **x** = 011101 ที่ปลายทางได้รับ จะอยู่ในทรงกลม S(010101,1) เท่านั้น และไม่อยู่ในทรงกลมอื่นใดเลย

ตัวอย่าง 1.11.3 : พิจารณารหัส C_3 = {000000, 010101, 101010, 111111} สมมุติว่า 010101 เป็นคำรหัสที่ส่ง และสมมุติว่ามีข้อผิดพลาดเกิด ขึ้น 2 ตำแหน่ง สมมุติว่ามีข้อผิดพลาดในตำแหน่งแรกและตำแหน่งที่ห้า ดังนั้นผู้รับจะได้รับ **x** = 110111 เปรียบเทียบระยะระหว่าง **x** กับคำรหัส **c** ทุกคำใน C_3 จะพบว่า d(**x**, **c**) = 1 มีค่า

ตาราง 1.11.2

ที่รับ **x**	คำรหัส **c**	d(**x**, **c**)
110111	000000	5
	010101	2
	101010	4
	111111	1

น้อยที่สุดเมื่อ **c** = 111111 ดังนั้น **c** = 111111 เป็นคำรหัสที่อยู่ใกล้ **x** ที่สุด ถ้าใช้หลักการถอดรหัสให้เป็นคำรหัสที่ใกล้ที่สุด เราจะสรุปว่า 111111 เป็นคำรหัสที่ส่งจากต้นทาง ซึ่งไม่ตรงกับคำรหัสที่ส่งจากต้นทาง แสดงว่าการถอดรหัสผิดพลาด จะเห็นว่าในกรณีนี้ 110111 ที่ปลายทางได้รับ จะอยู่ในทรงกลม S(111111, 1)

นิยาม 1.11.2 จะกล่าวว่ารหัส C สามารถแก้ข้อผิดพลาดได้ถึง t ตำแหน่ง ถ้ามีข้อผิดพลาดเกิดขึ้น t ตำแหน่งหรือน้อยกว่า คำที่รับจะอยู่ในทรงกลม S(**x**, t) เมื่อ **x** คือคำที่ส่ง แต่ไม่อยู่ใน S(**y**, t) สำหรับทุก ๆ **y** $\in$ C ซึ่ง **x** $\neq$ **y**

ทฤษฎีบท 1.11.2

> รหัส C สามารถแก้ไขข้อผิดพลาดได้ถึง t ตำแหน่ง ก็ต่อเมื่อ $d(C) \geq 2t + 1$

พิสูจน์ สมมุติให้ $d(C) \geq 2t + 1$ ให้ **x** คือคำรหัสที่ส่ง และให้ **y** เป็นคำที่ได้รับซึ่งผิดไปจาก **x** น้อยกว่าหรือเท่ากับ t ตำแหน่ง ให้ $\mathbf{x}' \in C$ ซึ่ง $\mathbf{x} \neq \mathbf{x}'$ จะเห็นว่า $d(\mathbf{y}, \mathbf{x}') > t$ เพราะถ้า $d(\mathbf{y}, \mathbf{x}') \leq t$ แล้ว

$$d(\mathbf{x}, \mathbf{x}') \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{x}') \leq t + t = 2t$$

ซึ่งเป็นไปไม่ได้ เพราะ $d(C) \geq 2t + 1$

ในทางกลับกัน สมมุติให้ C เป็นรหัสซึ่งสามารถแก้ข้อผิดพลาดได้ถึง t ตำแหน่ง จะแสดงว่า $d(C) \geq 2t + 1$ โดยใช้วิธีหาข้อขัดแย้ง โดยสมมุติว่า

$$d(C) < 2t + 1 \text{ หรือ } d(C) \leq 2t$$

แสดงว่าต้องมีคำรหัส **c** และ **d** ซึ่ง $d(\mathbf{c}, \mathbf{d}) = d(C) \leq 2t$ เราจะแสดงว่าเมื่อ **c** เป็นคำรหัสที่ส่งและเมื่อผิดไปไม่เกิน t ตำแหน่ง แล้วจะมีคำรหัสมากกว่าหนึ่งคำที่อยู่ใกล้คำที่รับ (ซึ่งอาจทำให้การถอดรหัสไม่ถูกต้องได้) หรือถอดรหัสคำที่รับผิดเป็น **d**

ก่อนอื่น เราสังเกตว่า $d(\mathbf{c}, \mathbf{d}) = d(C) \geq t + 1$ มิฉะนั้นแล้ว เมื่อส่งคำรหัส **c** ซึ่งผิดไปไม่เกิน t ตำแหน่ง แล้วอาจกลายเป็นคำรหัส **d** ได้ แสดงว่าการถอดรหัสนี้ผิดพลาด เพื่อความสะดวก เราจะให้

$$d(\mathbf{c}, \mathbf{d}) = d(C) = k$$

ดังนั้น $t + 1 \leq k \leq 2t$ และเพื่อความสะดวกในการอธิบาย เราจะสมมุติว่า **c** และ **d** แตกต่างกันใน $k = d(C)$ ตำแหน่งแรก เพราะถ้าไม่ใช่ เราสามารถสลับตำแหน่งของคำรหัสได้ พิจารณาเวกเตอร์ **x** ซึ่งมี $k - t$ ตำแหน่งแรกเหมือนกับ **c** และมี t ตำแหน่งถัดไปเหมือนกับ **d** และเหมือนกับทั้ง **c** และ **d** ใน $n - k$ ตำแหน่งสุดท้าย

$$\mathbf{x} = \underbrace{x_1x_2\cdots x_{k-t}}_{\text{เหมือน } \mathbf{c}}\ \underbrace{x_{k-t+1}\cdots x_k}_{\text{เหมือน } \mathbf{d}}\ \underbrace{x_{k+1}x_{k+2}\cdots x_n}_{\text{เหมือนทั้ง } \mathbf{c} \text{ และ } \mathbf{d}}$$

เนื่องจาก

$$d(\mathbf{c}, \mathbf{x}) = t \text{ และ } d(\mathbf{d}, \mathbf{x}) = k - t \leq t$$

ดังนั้น มีโอกาสเป็นไปได้ 2 กรณีคือ

$$d(\mathbf{c}, \mathbf{x}) = d(\mathbf{d}, \mathbf{x}) \text{ หรือ } d(\mathbf{c}, \mathbf{x}) \geq d(\mathbf{d}, \mathbf{x})$$

ถ้า $d(\mathbf{c}, \mathbf{x}) = d(\mathbf{d}, \mathbf{x})$ อาจทำให้ถอดรหัสผิดพลาด หรือถ้า $d(\mathbf{c}, \mathbf{x}) \geq d(\mathbf{d}, \mathbf{x})$ ซึ่งเมื่อใช้หลักการถอดรหัสให้เป็นคำที่ใกล้ที่สุด จะทำให้ถอดรหัส **x** ผิดเป็น **d** ■

ตัวอย่าง 1.11.4 : จากตัวอย่าง 1.9.8 เรามี $d(C_2) = 2$ และ $d(C_3) = 3$ ดังนั้น รหัส C_2 มีความสามารถในการตรวจจับข้อผิดพลาดได้ถึงหนึ่งตำแหน่ง แต่ไม่สามารถแก้ไขข้อผิดพลาดได้เลย ส่วนรหัส C_3 มีความสามารถในการตรวจจับข้อผิดพลาดได้ถึงสองตำแหน่งและสามารถแก้ไขข้อผิดพลาดได้ถึงหนึ่งตำแหน่ง

บทแทรก 1.11.1

1. ถ้า $d(C) = t$ รหัส C จะตรวจจับข้อผิดพลาดได้ถึง $t - 1$ ตำแหน่ง
2. ถ้า $d(C) = t$ รหัส C จะแก้ไขข้อผิดพลาดได้ถึง $\left\lfloor \frac{t-1}{2} \right\rfloor$ ตำแหน่ง เมื่อ $\lfloor x \rfloor$ คือจำนวนเต็มที่น้อยกว่าหรือเท่ากับ x

การสร้างรหัสแก้ไขข้อผิดพลาด จะยากกว่าการสร้างรหัสตรวจจับข้อผิดพลาด เราจะเลือกใช้รหัสแบบใดขึ้นอยู่กับว่า เราต้องการความน่าเชื่อถือมากน้อยเพียงใดในการสื่อสาร และต้องขึ้นอยู่กับระบบสื่อสารด้วย ในกรณีที่ระบบสื่อสารเป็นระบบสื่อสารแบบสองทาง และมีเวลาพอที่จะขอให้ต้นทางส่งข้อความมาใหม่ ถ้ารู้ว่าข้อ

ความที่ได้รับไม่ถูกต้อง ในกรณีนี้ เราอาจใช้รหัสชนิดที่ตรวจจับข้อผิดพลาดได้ก็พอ แต่บางครั้งเราก็ไม่มีทางเลือก เช่น กรณีที่ข้อมูลในเทปแม่เหล็กถูกทำลาย ไม่สามารถเรียกกลับคืนมาได้ ดังนั้น เราจำเป็นต้องใช้รหัสที่มีความสามารถในการแก้ไขข้อ ผิดพลาดได้

1.12 การสร้างรหัสใหม่จากรหัสเก่า

การสร้างรหัสใหม่จากรหัสที่มีอยู่แล้ว จะเป็นเครื่องมือที่มีประโยชน์ในกรณีที่เราต้องการหารหัสที่มีความยาว n และระยะน้อยสุด d ตามที่กำหนด

1.12.1 การขยายรหัส (Extending a Code)

เราได้เห็นการสร้างรหัสโดยการขยายคำรหัสให้ยาวขึ้นแล้วในตัวอย่าง 1.4.1 การขยายรหัสหมายถึงกระบวนการเพิ่มตำแหน่งในคำรหัสทุกคำ วิธีที่นิยมใช้กันมากวิธีหนึ่งคือวิธีที่เรียกว่าการเพิ่ม ***ตัวตรวจสอบภาวะเสมอ*** พิจารณากรณีที่ C เป็นรหัส ไบนารี-(n, M, d) เราจะสร้างรหัส $\hat{C}$ ที่ประกอบด้วยคำรหัส $\hat{c}$ ที่ได้จาก $c = c_1c_2 \ldots c_n$ ใน C โดยการเพิ่มบิตที่ n + 1 อีกหนึ่งบิตใน c ถ้า c มีน้ำหนักเป็นจำนวนคี่ เราเพิ่ม 1 แต่ถ้าน้ำหนักของ c เป็นจำนวนคู่ เราเพิ่ม 0 ถ้าให้

$$\hat{c} = c_1c_2 \ldots c_nc_{n+1}$$

เป็นผลลัพธ์ที่ได้จากการเพิ่มตัวตรวจสอบภาวะเสมอใน c เราได้

$$\hat{c} = \begin{cases} c_1c_2 \cdots c_n 1 & \text{เมื่อ wt(c) เป็นจำนวนคี่} \\ c_1c_2 \cdots c_n 0 & \text{เมื่อ wt(c) เป็นจำนวนคู่} \end{cases}$$

จะเห็นว่า แต่ละคำใน $\hat{C}$ มีน้ำหนักเป็นจำนวนคู่ และจากความรู้เรื่องจำนวนเต็มมอดุโล 2 (ดูรายละเอียดในบทที่ 2)

$$\sum_{i=1}^{n+1} c_i = 0 \pmod 2 \quad \text{หรือ} \quad c_{n+1} = \sum_{i=1}^{n} c_i \pmod 2$$

นั่นคือ

$$\hat{C} = \{\, c_1c_2 \ldots c_nc_{n+1} \in F_2^{n+1} \mid c_1c_2 \ldots c_n \in C,\ c_1+c_2+\ldots+c_n+c_{n+1} = 0\}$$

เรียกรหัสที่สร้างในลักษณะนี้ว่า ***รหัสตรวจสอบภาวะเสมอ***

ในกรณีทั่วไป ถ้า C เป็นรหัสบนฟิลด์จำกัด F_p เมื่อ p เป็นจำนวนเฉพาะ เราจะเพิ่มสัญลักษณ์เข้าไปในทุกคำรหัสของ C เพื่อทำให้ผลบวกของทุกตำแหน่งในคำรหัสเป็น 0 นั่นคือ ถ้า

$$\mathbf{c} = c_1c_2 \ldots c_n \text{ แล้ว } \hat{\mathbf{c}} = c_1c_2 \ldots c_nc_{n+1}$$

เมื่อ

$$\sum_{i=1}^{n+1} c_i \equiv 0 \pmod p \quad \text{หรือ} \quad c_{n+1} \equiv -\sum_{i=1}^{n} c_i \pmod p$$

ตัวอย่าง 1.12.1 : รหัส C_2 = {000, 011, 101, 110} ในตัวอย่าง 1.7.1 เป็นรหัสตรวจสอบภาวะเสมอ มีบิตที่สามเป็นบิตตรวจสอบภาวะเสมอ สำหรับคำรหัส $\mathbf{c} = c_1c_2c_3$ ใด ๆ ใน C_2 จะเห็นว่า

$$c_1 + c_2 + c_3 \equiv 0 \pmod 2 \quad \text{หรือ}$$

$$c_3 \equiv c_1 + c_2 \pmod 2$$

ทฤษฎีบท 1.12.1

สมมุติให้ d เป็นจำนวนเต็มคี่ ดังนั้น จะมีรหัสไบนารี-(n, M, d) ก็ต่อเมื่อมีรหัสไบนารี-(n+1, M, d+1)

พิสูจน์ : สมมุติให้ C เป็นรหัสไบนารี-(n, M, d) เมื่อ d เป็นจำนวนเต็มคี่ ให้ $\hat{C}$ เป็นรหัสไบนารีที่ประกอบด้วยคำรหัส $\hat{\mathbf{c}} = c_1c_2 \ldots c_nc_{n+1}$ ที่เกิดจากการเพิ่มบิต c_{n+1} ใน $\mathbf{c} = c_1c_2 \ldots c_n$ ดังนี้

$$\hat{\mathbf{c}} = \begin{cases} c_1c_2\cdots c_n1 & \text{เมื่อ } wt(\mathbf{c}) \text{ เป็นจำนวนคี่} \\ c_1c_2\cdots c_n0 & \text{เมื่อ } wt(\mathbf{c}) \text{ เป็นจำนวนคู่} \end{cases}$$

ให้ $\hat{\mathbf{x}}$ และ $\hat{\mathbf{y}}$ เป็นสมาชิกใน $\hat{C}$ ดังนั้น $wt(\hat{\mathbf{x}})$ และ $wt(\hat{\mathbf{y}})$ ต่างก็เป็นจำนวนเต็มคู่ จากทฤษฎีบท 1.9.2 เรามี

$$d(\hat{x},\hat{y}) = wt(\hat{x}) + wt(\hat{y}) - 2wt(\hat{x}\cap\hat{y})$$

ดังนั้น $d(\hat{x},\hat{y})$ ต้องเป็นจำนวนคู่ สำหรับทุกๆ $\hat{x}$, $\hat{y}$ ใน $\hat{C}$ นั่นคือ $d(\hat{C})$ ต้องเป็นจำนวนคู่ แต่

$$d \leq d(\hat{C}) \leq d + 1$$

และเนื่องจาก d เป็นจำนวนคี่ แสดงว่า $d(\hat{C}) = d + 1$

ในทางกลับกัน สมมุติให้ D เป็นรหัส-(n+1, M, d+1) แสดงว่าต้องมีคำรหัส **x**, **y** ใน D ซึ่ง

$$d(\mathbf{x}, \mathbf{y}) = d+1$$

เลือกตำแหน่งใน **x** และ **y** ที่แตกต่างกัน ตำแหน่งใดก็ได้ ลบตำแหน่งนั้นออกจากคำ รหัสทุกคำใน D จะทำให้ระยะน้อยสุดของ D ลดลงหนึ่ง ความยาวก็ลดลงหนึ่งด้วยเช่นกัน ส่วนจำนวนคำรหัสยังคงเดิม คือเท่ากับ M นั่นคือ เราได้รหัส-(n, M, d) ตาม ต้องการ ■

1.12.2 การทำให้รหัสสั้นลง(Shortening a Code)

สมมุติว่า C เป็นรหัส-(n, M, d) ฐาน q เราพิจารณาคำรหัสใน C ทั้งหลายที่มีสัญลักษณ์ตัวหนึ่งที่อยู่ในตำแหน่งที่เจาะจง เช่นพิจารณาคำรหัสใน C ที่มีสัญลักษณ์ λ อยู่ในตำแหน่งที่ j พิจารณาเฉพาะคำเหล่านี้ แล้วตัดสัญลักษณ์ λ ในตำแหน่งดังกล่าวออกจากทุกคำ เราจะได้รหัส C' ซึ่งมีความยาว $n - 1$ และระยะน้อยสุดของ C' จะมีค่าอย่างน้อย d เพราะว่าการทำให้ความยาวของรหัสสั้นลง อาจทำให้ระยะน้อยสุดเพิ่มขึ้น ซึ่งจะทำให้รหัสมีความสามารถในการแก้ไขข้อผิดพลาดได้ดีขึ้น แต่ข้อเสียคือจะทำให้จำนวนคำรหัสลดลง เรียกกระบวนการดังกล่าวนี้ว่า ***การตัดขวาง*** ที่ $c_j = \lambda$ เรียก C' ว่าเป็นรหัสที่เกิดจากการตัดขวางรหัส C ที่ $c_j = \lambda$

ตัวอย่าง 1.12.2 : พิจารณารหัส C = {0000, 0110, 0011, 1010, 1110}

จะพบว่า d(C) = 1 เราจะเลือกคำรหัสที่มี 0 ในตำแหน่งสุดท้าย แล้ว

ตัด 0 ในตำแหน่งสุดท้ายนี้ออก เราได้

ก่อนตัด	**หลังตัด**
0000	000
0110	011
1010	101
1110	111

จะเห็นว่า $C' = \{000, 011, 101, 111\}$ เป็นรหัสใหม่ที่ได้จากรหัส C โดยการตัดขวางรหัส C ที่ $c_4 = 0$ ความยาวของ C' เท่ากับ 3 นอกจากนี้ยังพบว่า $d(C') = 1$ แต่ถ้าเราให้ C'' เป็นรหัสที่เกิดจากการตัดขวางรหัส C ที่ $c_1 = 0$ เราได้

ก่อนตัด	**หลังตัด**
0000	000
0110	110
0011	011

จะเห็นว่า

$$C'' = \{000, 110, 011\}$$

ซึ่งมีความยาวเท่ากับ 3 เช่นกัน แต่ $d(C'') = 2$

1.13 รหัสที่สมมูลกัน (Equivalence of Codes)

ในหัวข้อนี้ เราจะเน้นศึกษารหัสที่ไม่เหมือนกัน แต่มีคุณลักษณะเหมือนกัน นั่นคือมีตัวแปร n, M และ d เหมือนกัน ก่อนอื่นเราทบทวนความรู้เรื่องการเรียงสับเปลี่ยนของเซต

$$S = \{ a_1, a_2, \ldots, a_n \}$$

ที่มีขนาด n การเรียงสับเปลี่ยนของเซต S ก็คือฟังก์ชัน 1-1 จากเซต S ไปทั่วถึงเซต S ถ้า f เป็นการเรียงสับเปลี่ยนของเซต S เราจะเขียน

$$\begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ f(a_1) & f(a_2) & \cdots & f(a_n) \end{pmatrix}$$

แทนฟังก์ชัน f ที่ส่ง a_1 ไป $f(a_1)$, ส่ง a_2 ไป $f(a_2)$, ..., และส่ง a_n ไป $f(a_n)$ เช่น

$$\begin{pmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \end{pmatrix}$$

เป็นฟังก์ชันที่ส่ง 0 ไป 2, ส่ง 1 ไป 0, และส่ง 2 ไป 1 เป็นต้น

นิยาม 1.13.1 จะกล่าวว่ารหัสฐาน q สองรหัสสมมูลกัน ถ้ารหัสหนึ่งได้จากอีกรหัสหนึ่ง โดยการดำเนินการต่อไปนี้อย่างใดอย่างหนึ่ง หรือผสมผสานกัน

1. การเรียงสับเปลี่ยนตำแหน่งของคำรหัส
2. การเรียงสับเปลี่ยนสัญลักษณ์ที่ปรากฏในตำแหน่งที่ i ของทุกคำรหัส

ตัวอย่าง 1.13.1 : ให้ C_4 = {012210, 112112, 221020} เป็นรหัสเทอร์นารี-(6, 3, 3) ในตัวอย่าง 1.9.9 ถ้าเราสลับสัญลักษณ์ที่อยู่ในตำแหน่งแรกกับตำแหน่งสุดท้ายของคำรหัสทุกคำใน C_4 กล่าวคือ เราใช้การเรียงสับเปลี่ยน

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 2 & 3 & 4 & 5 & 1 \end{pmatrix}$$

กับตำแหน่งของคำรหัสใน C_4 เราได้รหัสใหม่

$$D_1 = \{012210, 212111, 021022\}$$

จะเห็นว่าคำรหัสใน D_1 ไม่เหมือนกับคำรหัสใน C_4 ยกเว้นคำแรก แต่รหัส D_1 มีความยาวเท่ากับ 6 และมีระยะน้อยสุดเท่ากับ 3 เช่นเดิม

ตัวอย่าง 1.13.2 : พิจารณารหัส C_4 ในตัวอย่าง 1.9.9 เช่นเดิม ถ้าเราเรียงสับเปลี่ยนสัญลักษณ์ในตำแหน่งแรกของคำรหัสทุกคำใน C_4 โดยใช้การเรียงสับเปลี่ยน

$$\begin{pmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \end{pmatrix}$$

นั่นคือเปลี่ยนสัญลักษณ์ในตำแหน่งแรกของแต่ละคำจาก 0 เป็น 2 เปลี่ยน 2 เป็น 1 และเปลี่ยน 1 เป็น 0 เราได้รหัสใหม่

$$D_2 = \{212210, 012112, 121020\}$$

จะเห็นว่าการกระทำเช่นนี้ไม่ทำให้ค่า d ซึ่งเป็นระยะน้อยสุดเปลี่ยนแปลงไป ทั้งนี้เพราะว่าระยะระหว่างสองคำรหัสใด ๆ ใน D_2 ยังคงเหมือนกับใน C_4 ตำแหน่งที่แตกต่างกันก็ยังคงแตกต่างกัน และตำแหน่งที่เหมือนกันก็ยังคงเหมือนกันเช่นเดิม

ทฤษฎีบทประกอบ 1.13.1

> รหัสที่สมมูลกันจะมีตัวแปร (n, M, d) เหมือนกัน

ทฤษฎีบทประกอบ 1.13.2

> ถ้าชุดตัวอักษร A มี 0 เป็นสมาชิกแล้ว รหัสใด ๆ บนเซต A จะสมมูลกับรหัสที่มี **0** = 00 … 0 เป็นคำรหัส

ตัวอย่าง 1.13.3 : พิจารณารหัสไบนารี C = {0010, 0011, 1011, 1111} ซึ่งมีความยาวเท่ากับ 4 มี d = 1 สมมุติว่าเราต้องการหารหัส D ซึ่งสมมูลกับรหัส C โดยที่มีคำรหัสศูนย์ **0** = 0000 อยู่ใน D

เราทำได้โดยการใช้การเรียงสับเปลี่ยน $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ กับตำแหน่งที่สามของคำรหัสทุกคำใน C เราได้

C		D
0010		0000
0011	→	0001
1011		1001
1111		1101

ตัวอย่าง 1.13.4 : พิจารณารหัส C_4 = {012210, 112112, 221020} ในตัวอย่าง 1.13.1 ถ้าใช้การเรียงสับเปลี่ยน

$$\begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 2 \end{pmatrix}$$

กับสัญลักษณ์ในตำแหน่งที่สองและตำแหน่งที่ห้า เราได้รหัส D_1 และใช้

$$\begin{pmatrix} 0 & 1 & 2 \\ 2 & 1 & 0 \end{pmatrix}$$

กับสัญลักษณ์ในตำแหน่งที่สามและตำแหน่งที่สี่ เราได้รหัส D_2

C_4		D_1		D_2
012210		002200		000000
112112	$\longrightarrow$	102102	$\longrightarrow$	100102
221020		221020		221220

ซึ่งสมมูลกับรหัส C_4 และมีคำรหัสศูนย์ 000000 อยู่ใน D_2

1.14 รหัสสมบูรณ์ (Perfect Code)

พิจารณาอสมการทั้งสองในทฤษฎีบท 1.9.1 รหัสใดที่มีสมบัติทำให้จำนวนทางซ้าย เท่ากับจำนวนทางขวาของอสมการแล้ว เราจะเรียกรหัสนั้นว่า**รหัสสมบูรณ์** นั่นคือถ้า C เป็นรหัส-(n, M, 2t+1) ฐาน q ใด ๆ ซึ่ง

$$M\left\{\binom{n}{0}+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\ldots+\binom{n}{t}(q-1)^t\right\}=q^n \quad \ldots(1.14.1)$$

หรือ

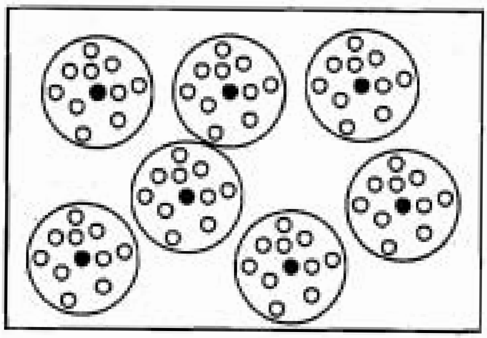
$$M\left\{\binom{n}{0}+\binom{n}{1}+\binom{n}{2}+\ldots+\binom{n}{t}\right\}=2^n \text{ สำหรับ } q=2 \quad \ldots(1.14.2)$$

แล้ว C จะเป็นรหัสสมบูรณ์ สำหรับรหัส-(n, M, 2t+1) ฐาน q จะเห็นว่าจำนวน

$$\binom{n}{0}+\binom{n}{1}(q-1)+\binom{n}{2}(q-1)^2+\ldots+\binom{n}{t}(q-1)^t$$

ที่อยู่ในวงเล็บปีกกาของอสมการ 1.14.1 และ 1.14.2 ก็คือจำนวนสมาชิกใน F_q^n ที่อยู่ในทรงกลม S(c, t) สำหรับ c ที่เป็นคำรหัสใด ๆ ใน C นั่นเอง ดังนั้น ถ้าเรานำทรงกลมทั้งหลายซึ่งมีทั้งหมด M ทรงกลมนำมายูเนียนกัน เราจะได้

$$\bigcup_{c \in C} S(\mathbf{c}, t) = F_q^n$$



แสดงว่าสมาชิกแต่ละตัวใน F_q^n ต้องอยู่ในทรงกลมใดทรงกลมหนึ่ง นั่นคืออยู่ในรัศมี t จากจุดศูนย์กลางของทรงกลม กล่าวคือ อยู่ในรัศมี t ของคำรหัสบางคำ

ตัวอย่าง 1.14.1 : พิจารณารหัส-(n, q^n, 1) ฐาน q จะเห็นว่า t = 0 และรหัสนี้ประกอบด้วยสมาชิกทุกตัวใน F_q^n เมื่อแทนตัวแปรลงในสมการ (1.14.1) จะพบว่าสอดคล้องกับสมการดังกล่าว ดังนั้น F_q^n เป็นรหัสสมบูรณ์

ตัวอย่าง 1.14.2 : พิจารณารหัสฐานไบนารีที่มีความยาว n และมีสมาชิกตัวเดียว เช่น C = {0} จะเห็นว่าระยะน้อยสุดของ C ไม่นิยาม เพราะมีเพียงคำรหัสเดียว แต่ถ้าเราให้ระยะน้อยสุดของ C คือ d = 2n+1 จะพบว่าตัวแปรของรหัส C สอดคล้องกับสมการ (1.14.2) ดังนั้น C เป็นรหัสสมบูรณ์

ตัวอย่าง 1.14.3 : พิจารณารหัสไบนารี-(2t + 1, 2, 2t +1) แบบซ้ำ ในที่นี้ n = 2t + 1 ซึ่งเป็นจำนวนคี่ และ M = 2 แทนค่าตัวแปรใน (1.14.2) เราได้

$$2\left\{\binom{2t+1}{0}+\binom{2t+1}{1}+\binom{2t+1}{2}+\ldots+\binom{2t+1}{t}\right\}$$

$= 2 \times 2^{2t} = 2^{2t+1} = 2^{n}$

ดังนั้น รหัสไบนารี-(2t+1, 2, 2t+1) แบบซ้ำ ที่มีความยาวเป็นจำนวนคี่ เป็นรหัสสมบูรณ์

ตัวอย่าง 1.14.4 :

1. รหัสไบนารี-(7, 2^4, 3) เป็นรหัสสมบูรณ์ เพราะ

$$2^4\left\{\binom{7}{0}+\binom{7}{1}\right\} = 2^4\{1+7\} = 2^7$$

2. รหัสโกเลย์ไบนารี (binary Golay code) คือรหัสไบนารี-(23, 2^{12}, 7) เป็นรหัสสมบูรณ์ (ดูรายละเอียดในหัวข้อ 4.2) เพราะ

$$2^{12}\left\{\binom{23}{0}+\binom{23}{1}+\binom{23}{2}+\binom{23}{3}\right\}$$

$$= 2^{12}\{1+23+253+1771\} = 2^{12} \times 2^{11} = 2^{23}$$

3. รหัสโกเลย์เทอร์นารี (ternary Golay code) คือรหัสเทอร์นารี-(11, 3^6, 5) เป็นรหัสสมบูรณ์ (ดูรายละเอียดในหัวข้อ 4.2) เพราะ

$$3^6\left\{\binom{11}{0}+\binom{11}{1}2+\binom{11}{2}2^2\right\}$$

$$= 3^6\{1+22+220\} = 3^6 \times 3^5 = 3^{11}$$

1.15 รหัสมากสุด (Optimal Code)

สำหรับค่า n และ d ที่กำหนดให้ เราให้ $A_q(n, d)$ แทนค่า M ที่มากที่สุดในระหว่างรหัสฐาน q ทั้งหลาย ที่เป็นรหัสที่มีตัวแปร (n, M, d) กล่าวคือ

$A_q(n, d) = \max(M \mid$ มีรหัส-(n, M, d) ฐาน q$)$

จำนวน $A_q(n, d)$ มีบทบาทสำคัญต่อวิชาทฤษฎีรหัส การหาค่าของ $A_q(n, d)$ เป็นหัวข้อวิจัยที่นักคณิตศาสตร์จำนวนมากนิยมศึกษาและค้นคว้าขยายผล ในหัวข้อนี้เราจะแสดงการหาค่าของ $A_q(n, d)$ สำหรับค่า n และ d ที่มีค่าเล็ก ๆ บางค่า

ในการหาค่าของ $A_q(n, d)$ เราจะใช้ประโยชน์จากความรู้ในหัวข้อ 1.13 ที่กล่าวว่า สำหรับรหัส C ใด ๆ เราสามารถหารหัส C' ซึ่งสมมูลกับรหัส C และ C' มีเวกเตอร์ศูนย์เป็นคำรหัสใน C' ถ้าชุดอักษรของรหัสไม่มีสัญลักษณ์ 0 ให้เลือกสัญลักษณ์ใดสัญลักษณ์หนึ่งแล้วแทนสัญลักษณ์นั้นด้วย 0

ทฤษฎีบท 1.15.1

1. $A_q(n, 1) = q^n$
2. $A_q(n, n) = q$

พิสูจน์ 1. เนื่องจากรหัสที่มีระยะน้อยสุดเท่ากับ d = 1 คำรหัสแต่ละคู่มีตำแหน่งที่แตกต่างกันเพียงตำแหน่งเดียวก็พอ ดังนั้นทุกคำใน F_q^n เป็นคำรหัส นั่นคือ $A_q(n, 1) = q^n$ เพราะ $|F_q^n| = q^n$

พิสูจน์ 2. สมมุติให้ C เป็นรหัส-(n, M, n) ฐาน q เนื่องจาก d(C) = n แสดงว่าตำแหน่งที่ 1 ของคำรหัสแต่ละคำใน C จะต้องแตกต่างกัน แสดงว่า $M \leq q$ ดังนั้น $A_q(n, n) \leq q$ ในทางกลับกัน เรารู้ว่ารหัสฐาน q แบบซ้ำและมีความยาว n มีคำรหัสทั้งหมด q คำ นั่นคือ $A_q(n, n) = q$ ∎

ทฤษฎีบท 1.15.2

$A_2(4, 3) = 2$

พิสูจน์ ให้ C เป็นรหัส-(4, M, 3) เราจะคิดว่า C เป็นรหัสซึ่งมีเวกเตอร์ศูนย์ **0** = 0000 เป็นคำรหัสหนึ่งใน C และเนื่องจาก d(C) = 3 ดังนั้น

คำรหัส c ใด ๆ ใน C จะต้องมี $d(c, 0) \geq 3$ แสดงว่าจะต้องมี 1 ใน c อย่างน้อยสามตำแหน่ง นั่นคือ คำที่อยู่ใน C จะต้องเป็นคำในห้าคำต่อไปนี้

1110, 1101, 1011, 0111, หรือ 1111

จะเห็นว่าคำแต่ละคู่ในห้าคำนี้มีระยะห่างกันน้อยกว่า 3 ดังนั้น คำเหล่านี้จะอยู่ใน C ได้เพียงคำเดียวเท่านั้น ถ้าอยู่มากกว่าหนึ่งคำ จะทำให้ $d(C) < 3$ แสดงว่า $A_2(4, 3) \leq 2$ นอกจากนี้

$$C = \{0000, 1110\}$$

เป็นรหัส-(4, 2, 3) เราได้ $A_2(4, 3) \geq 2$ ดังนั้น $A_2(4, 3) = 2$ ■

ทฤษฎีบท 1.15.3

$A_2(5, 3) = 4$

พิสูจน์ ให้ C เป็นรหัส - (5, M, 3) พิจารณารหัส C_0 ซึ่งเป็นรหัสตัดขวางของ C ที่ $c_1 = 0$ ให้ d_0 และ M_0 เป็นระยะน้อยสุดและขนาดของ C_0 ตามลำดับ เราได้ $d_0 \geq 3$ และเนื่องจากเรารู้ว่า $A_2(4, 3) = 2$ และ $A_2(4, 4) = 2$ แสดงว่า $M_0 \leq 2$ ในทำนองเดียวกัน ถ้าให้ C_1 เป็นรหัสตัดขวางของ C ที่ $c_1 = 1$ เราจะได้ $M_1 \leq 2$ เมื่อ M_1 คือขนาดของ C_1 ดังนั้น $M = M_0 + M_1 \leq 4$ นั่นคือ $A_2(5, 3) \leq 4$ ในทางกลับกัน เราสามารถตรวจสอบได้ไม่ยากนักว่า

$$\{00000, 11100, 00111, 11011\}$$

เป็นรหัส-(5, 4, 3) แสดงว่า $A_2(5, 3) \geq 4$ ดังนั้น $A_2(5, 3) = 4$ ■

จะเห็นว่า วิธีการที่เราใช้ในการพิสูจน์ทฤษฎีบท 1.15.3 เหมาะสำหรับกรณีที่ n และ d มีค่าเล็ก ๆ สำหรับกรณีที่ n และ d มีขนาดใหญ่ ๆ แล้ว การคำนวณค่าของ $A_q(n, d)$ จะต้องใช้วิธีการที่ซับซ้อนกว่านี้มาก ค่าของ $A_q(n, d)$ ที่คำนวณได้แล้วมีจำนวนไม่มาก ตาราง 1.15.1 สรุปค่าของ $A_2(n, d)$ สำหรับ $n \leq 16$

ตาราง 1.15.1: แสดงค่าของ $A_2(n, d)$

n	d = 3	d = 5	d = 7
5	4	2	-
6	8	2	-
7	16	2	2
8	20	4	2
9	40	6	2
10	72 – 79	12	2
11	144 – 158	24	4
12	256	32	4
13	512	64	8
14	1024	128	16
15	2048	256	32
16	2560 – 3276	256 – 340	36 – 37

และ $d \le 7$ ที่คำนวณได้และเป็นที่รู้กันแล้ว ตารางนี้คัดลอกมาจากหนังสือของ Sloane (1982) หน้า 156 ในกรณีที่ยังหาค่าของ $A_2(n, d)$ ที่แน่นอนไม่ได้ หาได้เพียงขอบเขตบนและขอบเขตล่างของ $A_2(n, d)$ เท่านั้น เช่นในกรณีที่ $n = 10$, $d = 3$ ขอบเขตล่างและขอบเขตบนของ $A_2(n, d)$ คือ 72 และ 79 ตามลำดับ นั่นคือ

$$72 \le A_2(n, d) \le 79$$

เป็นต้น

ทฤษฎีบท 1.15.4

The sphere-covering bound for $A_q(n, d)$

ถ้า $V_q^n(d-1)$ แทนขนาดของทรงกลมใน F_q^n ที่มีรัศมี $d - 1$ แล้ว

$$\frac{q^n}{V_q^n(d-1)} \le A_q(n, d)$$

พิสูจน์ สมมุติให้ $C = \{c_1, c_2, \dots, c_M\}$ เป็นรหัส-(n, M, d) ซึ่งเป็นรหัสมากสุดฐาน q นั่นคือเป็นรหัสฐาน q ที่มีจำนวนคำรหัสมากที่สุด ดังนั้น $M = A_q(n, d)$ และไม่มีเวกเตอร์ใดใน F_q^n ที่ห่างจากคำรหัสแต่ละคำ

เป็นระยะมากกว่าหรือเท่ากับ d เพราะถ้ามีเวกเตอร์ดังกล่าวแล้ว เราสามารถเพิ่มเวกเตอร์นั้นเข้าไปใน C ซึ่งจะทำให้เราได้รหัส-(n,M+1, d) ซึ่งเป็นไปไม่ได้ เพราะ C เป็นรหัสขนาด M ซึ่ง $M = A_q(n, d)$ ดังนั้นทุกเวกเตอร์ใน F_q^n ต้องอยู่ห่างจากคำรหัสแต่ละคำเป็นระยะมากที่สุด d − 1 แสดงว่าเวกเตอร์แต่ละเวกเตอร์ใน F_q^n ต้องเป็นสมาชิกในทรงกลม $S(\mathbf{c}_i, d-1)$ สำหรับบางค่าของ i ซึ่ง $i = 1, 2, \ldots, M$ ในกรณีนี้เรากล่าวว่าเซต

$$\{S(\mathbf{c}_i, d-1) \mid i = 1, 2, \ldots, M\}$$

ของทรงกลมทั้งหลายคลุมเซต F_q^n กล่าวคือ

$$F_q^n \subset \bigcup_{i=1}^{M} S(\mathbf{c}_i, d-1)$$

เมื่อคิดถึงขนาดของเซต และเนื่องจาก $|F_q^n| = q^n$ เราได้

$$\begin{aligned} q^n &\le \left| \bigcup_{i=1}^{M} S(\mathbf{c}_i, d-1) \right| \\ &\le \sum_{i=1}^{M} |S(\mathbf{c}_i, d-1)| \\ &= \sum_{i=1}^{M} V_q^n(d-1) = V_q^n(d-1) \cdot M \end{aligned}$$

เมื่อ $V_q^n(d-1)$ แทนจำนวนสมาชิกในทรงกลม $S(\mathbf{c}, d-1)$ สำหรับ $\mathbf{c}$ ใด ๆ ใน F_q^n ดังนั้น

$$\frac{q^n}{V_q^n(d-1)} \le M = A_q(n, d)$$

ตามต้องการ ∎

บทแทรก 1.15.1

$$\boxed{\frac{q^n}{V_q^n(d-1)} \le A_q(n, d) \le \frac{q^n}{V_q^n\left(\left\lfloor \frac{d-1}{2} \right\rfloor\right)}}$$

พิสูจน์ เป็นผลโดยตรงจากจาก sphere-packing bound (ทฤษฎีบท 1.9.1) และ sphere-covering bound (ทฤษฎีบท 1.15.4) ∎

ตัวอย่าง 1.15.1 : เราจะแสดงว่า $A_2(5, 4) = 2$ โดยใช้ sphere-covering bound

จาก sphere-covering bound เราได้

$$A_2(5, 4) \geq \frac{2^5}{V_2^5(3)} = \frac{32}{\binom{5}{0}+\binom{5}{1}+\binom{5}{2}+\binom{5}{3}} = \frac{32}{26}$$

ดังนั้น $A_2(5, 4) \geq 2$ และจากทฤษฎีบท 1.12.1 เราได้

$$A_2(5, 4) = A_2(4, 3)$$

และจากทฤษฎีบท 1.9.1 เราได้

$$A_2(4, 3) \leq \frac{2^4}{V_2^5(1)} = \frac{16}{\binom{5}{0}+\binom{5}{1}} = \frac{16}{6}$$

นั่นคือ $A_2(4, 3) \leq 2$ ดังนั้น เราสามารถสรุปได้ว่า

$$A_2(5, 4) = A_2(4, 3) = 2$$

ทฤษฎีบท 1.15.5

The Singleton bound

$$A_q(n, d) \leq q^{n-d+1}$$

พิสูจน์ ให้ C เป็นรหัส-(n, M, d) พิจารณาคำรหัสใน C ถ้าเราลบ d – 1 ตำแหน่งสุดท้ายของแต่ละคำออก ส่วนที่เหลือจะเป็นเวกเตอร์ที่มีความยาว n – d + 1 และเวกเตอร์เหล่านี้ต้องแตกต่างกันทั้งหมด เพราะถ้ามีเวกเตอร์ส่วนที่เหลือคู่ใดเหมือนกัน แสดงว่าจำนวนตำแหน่งที่แตกต่างกันของเวกเตอร์คู่นั้น ต้องน้อยกว่าหรือเท่ากับ d – 1 ตำแหน่ง ดังนั้น จำนวนคำรหัสใน C ต้องมีจำนวนไม่น้อยกว่าจำนวนเวกเตอร์ที่เหลือจากการตัด แต่จำนวนเวกเตอร์ที่เหลือจากการตัดซึ่งแตกต่างกันนั้น มีจำนวนเท่ากับ q^{n-d+1} ดังนั้น

$$M = A_q(n, d) \leq q^{n-d+1}$$

ตามต้องการ ■

ตัวอย่าง 1.15.2 : จาก The Singleton bound เราได้

$$A_q(4, 3) \le q^2$$

แต่จาก sphere-packing bound เรามี

$$A_q(4, 3) \le \frac{q^4}{\binom{4}{0} + \binom{4}{1}(q-1)} = \frac{q^4}{4q-3}$$

จะเห็นว่าถ้า $q \ge 4$ แล้ว Singleton bound จะดีกว่า sphere-packing bound มาก เพราะ $q^2 < \frac{q^4}{4q-3}$

แบบฝึกหัด 1

1. สมมุติว่าเราใช้รหัสไบนารีแบบซ้ำที่มีความยาว 5 บนช่องสัญญาณ BSC ซึ่งมี p เป็นความน่าจะเป็นไขว้ จงแสดงว่าความน่าจะเป็นที่จะถอดคำรหัสผิดพลาด(word error probability) คือ $10p^3-15p^4+6p^5$
2. ถ้าเป็นไปได้ จงสร้างรหัสไบนารี-(n, M, d) ที่มีตัวแปรต่อไปนี้ (6, 2, 6), (3, 8, 1), (4, 8, 2), (5, 3, 4), (8, 30, 3) ถ้าเป็นไปไม่ได้ จงให้เหตุผล
3. พิจารณารหัส C ที่ประกอบด้วยสมาชิกทั้งหมดใน F_2^n ที่มีเลข 1 เป็นจำนวนคู่ จงหาความยาว ขนาด และระยะน้อยสุดของ C
4. ให้ E_n เป็นเซตของเวกเตอร์ใน F_2^n ที่มีน้ำหนักเป็นจำนวนคู่ จงแสดงว่า E_n เป็นรหัสที่ได้จากรหัส F_2^{n-1} การเพิ่มบิตตรวจสอบภาวะเสมอใน แสดงว่า E_n เป็นรหัส-(n, 2^{n-1}, 2)
5. สมมุติว่าคำรหัสจากรหัส {000, 111} ถูกส่งผ่านช่องสัญญาณ BSC ที่มี p = 0.01 เป็นความน่าจะเป็นไขว้(ความน่าจะเป็นที่สัญลักษณ์จะผิด) จงถอดรหัสคำต่อไปนี้โดยใช้หลักความน่าจะเป็นสูงสุด

   5.1 010 5.2 011 5.3 001

6. สมมุติว่าคำรหัสจากรหัส {000, 100,111} ถูกส่งผ่านช่องสัญญาณ BSC ที่มี p = 0.01 เป็นความน่าจะเป็นไข้ว จงถอดรหัสคำต่อไปนี้โดยใช้หลักความน่าจะเป็นสูงสุด

   6.1 010 6.2 011 6.3 001

7. จงหาระยะแฮมมิ่งต่อไปนี้

   7.1 d(0011101, 1110000) 7.2 d(122101, 001221)

8. สำหรับ C = {01101, 00011, 10110, 11000} จงถอดรหัสคำต่อไปนี้โดยใช้หลักการถอดรหัสให้เป็นคำรหัสที่ใกล้ที่สุด

   8.1 00000 8.2 10110 8.3 11011

9. สำหรับ C = {012210, 112112, 221020} จงถอดรหัสคำต่อไปนี้โดยใช้หลักการถอดรหัสให้เป็นคำรหัสที่ใกล้ที่สุด

   9.1 011221 9.2 112000 9.3 120120

10. ถ้า 001 เป็นคำที่ได้รับ จงอธิบายการถอดรหัสแบบ IMLD สำหรับรหัสต่อไปนี้

    10.1 C = {101, 110, 111}

    10.2 C = {000, 100, 010, 011}

11. จงหาระยะน้อยสุดของรหัสต่อไปนี้

    11.1 C = {11100, 01001, 10010, 00111}

    11.2 C = {120, 110, 201, 220}

12. จงหาทรงกลมต่อไปนี้

    12.1 S(11011, 1) ใน F_2^5 12.2 S(11011, 2) ใน F_2^5

    12.3 S(11011, 1) ใน F_3^5 12.4 S(11011, 2) ใน F_3^5

13. จงขยายรหัส C = {00000, 11100, 00111, 11011} โดยการเพิ่มบิตตรวจสอบภาวะเสมอ และหาตัวแปรของรหัส

14. กำหนดให้

$C = \{0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111,$
$1110, 1111\}$

จงหารหัสที่เกิดจากการตัดขวาง $c_2 = 1$ ของ C

15. จงหารหัสไบนารีที่มีตัวแปร $(5, 4, 3)$

16. จงแสดงว่า $A_2(5, 3) = 4$

17. จงหาความน่าจะเป็นที่จะถอดคำรหัสผิดพลาด $P_{err}(C)$ เมื่อ $C = \{000, 100, 110\}$