นิยามใหม่ของรีดักโดยใช้ทฤษฎีฟังก์ชันการขึ้นต่อกัน

A NEW DEFINITION OF REDUCT BASED ON FUNCTIONAL DEPENDENCY THEORY

ดวงใจ วรเศรษฐเมธี
DUANGJAI VORASATMATEE

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2549

ISBN 974-15-2128-6

นิยามใหม่ของรีดักโดยใช้ทฤษฎีฟังก์ชันการขึ้นต่อกัน

A NEW DEFINITION OF REDUCT BASED ON FUNCTIONAL DEPENDENCY THEORY

ดวงใจ วรเศรษฐเมธี

DUANGJAI VORASATMATEE

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2549

ISBN 974-15-2128-6

A NEW DEFINITION OF REDUCT BASED ON FUNCTIONAL DEPENDENCY THEORY

DUANGJAI VORASATMATEE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE IN COMPUTER SCIENCE

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2006

ISBN 974-15-2128-6

หัวข้อวิทยานิพนธ์	นิยามใหม่ของรีดักโดยใช้ทฤษฎีฟังก์ชันการขึ้นต่อกัน
นักศึกษา	นางสาวดวงใจ วรเศรษฐเมธี
รหัสประจำตัว	45064613
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์
พ.ศ.	2549
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.วีระ บุญจริง

บทคัดย่อ

งานวิจัยนี้เสนอนิยามใหม่ของรีดักโดยนิยามบนนิยามฟังก์ชันการขึ้นต่อกัน นิยามใหม่นี้มีการนิยามให้สมมูลกับนิยามของรีดักตามทฤษฎีรัฟเซตเพื่อให้สามารถใช้ประโยชน์จากทฤษฎีต่างๆ ของฟังก์ชันการขึ้นต่อกัน จากนั้นงานวิจัยนี้เสนออัลกอริทึมค้นหาใหม่สำหรับค้นหารีดักตามนิยามใหม่นี้ อัลกอริทึมนี้ค้นหาคำตอบจากโครงข่ายคำตอบอย่างมีประสิทธิภาพโดยการตัดคำตอบที่เป็นไปได้ในโครงข่ายคำตอบด้วยกฎต่างๆ ที่ได้จากนิยามและคุณสมบัติฟังก์ชันการขึ้นต่อกัน กฎเกณฑ์ที่ใช้รับประกันความครบถ้วนและถูกต้องของคำตอบ

Thesis Title	A New Definition of Reduct based on Functional Dependency Theory
Student	Miss. Duangjai Vorasatmatee
Student ID.	45064613
Degree	Master of Science
Programme	Computer Science
Year	2006
Thesis Advisor	Assoc.Prof.Dr.Veera Boonjing

ABSTRACT

This research proposes a new definition of reduct based on functional dependency definition. The new definition is defined to be equivalent to the definition of reduct in rough set theory to allow uses of functional dependency theories. The research then proposes a new search algorithm for finding reducts based on this new definition. The algorithm searches for all reducts on solution lattice in an efficient way by pruning possible solutions on the lattice using rules derived from the new definition and properties of functional dependencies. The rules used guarantee completeness and optimality of solutions.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ คำปรึกษา ความรู้ ความดูแลเอาใจใส่และหนังสือต่างๆ จาก รศ.ดร. วีระ บุญจริง ผู้เป็นอาจารย์ที่ปรึกษา ซึ่งท่านได้สละเวลาให้กับข้าพเจ้าอย่างเต็มที่ ข้าพเจ้ารู้สึกซาบซึ้งเป็นอย่างยิ่ง จึงใคร่ขอกราบขอบพระคุณเป็นอย่างสูงยิ่ง

ขอกราบขอบพระคุณ ผศ.ดร. ศรัณย์ อินทโกสุม และ ผศ.ดร.จีรพร ศรีสวัสดิ์ สำหรับคำแนะนำ และคำปรึกษาต่าง ๆ ในการทำวิทยานิพนธ์ด้วยดีมาโดยตลอด

ขอกราบขอบพระคุณ ดร.เฉลิมศักดิ์ เลิศวงศ์เสถียร ซึ่งให้ความกรุณามาเป็นตัวแทนกรรมการจากบุคคลภายนอกทำให้ได้รับคำปรึกษาและคำแนะนำต่าง ๆ ในการสอบวิทยานิพนธ์

ขอกราบขอบพระคุณ Jianchao Han ซึ่งให้เอกสารบทความแก่ข้าพเจ้า ทำให้ข้าพเจ้าสามารถศึกษางานวิจัยที่เกี่ยวข้องได้เข้าใจดียิ่งขึ้น

ขอกราบขอบพระคุณคณาจารย์ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ต่าง ๆ ให้กับข้าพเจ้าสำหรับการเรียนในระดับการศึกษาปริญญาโท

ขอขอบพระคุณ อาแหมะ ยี่โกและพี่หวีที่ช่วยดูแลน้องหลินหลงและงานในบ้านทำให้ข้าพเจ้ามีเวลาในการเรียนและทำวิทยานิพนธ์ได้มากขึ้น

ขอขอบพระคุณ ตา ยาย ป้า น้า อา ทุกคน ที่คอยให้คำแนะนำและตักเตือน รวมทั้งเป็นกำลังใจด้วยดีตลอดมา

ขอขอบคุณ ต่าย นุช หญิงและเพื่อนๆ พี่ น้องๆ ปริญญาโททุกคน ที่ให้คำแนะนำและให้ความช่วยเหลือเป็นอย่างดีเสมอมาทั้งการเรียนปริญญาโทและการทำวิทยานิพนธ์

ขอขอบคุณ น้องหยก อมยิ้ม ความ เฟรน พี่เอก พี่ตูน รีฟิล ตั้ม เพื่อนใจ แพน โป เมธและเพื่อนๆ ทุกคนที่คอยให้กำลังใจในการทำวิทยานิพนธ์อย่างดีตลอดมา

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา ที่ให้การสนับสนุนในการเรียนเป็นอย่างดีตลอดมา ทำให้ข้าพเจ้ามีกำลังใจพยายามตั้งใจเรียนในระดับการศึกษาปริญญาโทและการทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

สำหรับคุณงามความดีและประโยชน์อันพึงมาจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน บิดา มารดา อาจารย์ทุก ๆ ท่านซึ่งเป็นที่เคารพรักยิ่ง ตลอดจนญาติพี่น้องและเพื่อน ๆ ทุกคน

ดวงใจ วรเศรษฐเมธี

สารบัญ

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญตาราง

สารบัญตาราง(ต่อ)

สารบัญรูป

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การเลือกลักษณะเป็นขั้นตอนหนึ่งที่สำคัญของกระบวนการทำเหมืองข้อมูลเพื่อค้นหาเซตย่อยลักษณะที่เหมาะสมที่สุด [3] ซึ่งการทำเหมืองข้อมูลเกี่ยวข้องกับการวิเคราะห์ข้อมูลจำนวนมหาศาลและมีความซับซ้อนมาก จึงกลายเป็นงานที่มีความสำคัญมากต่อการบ่งชี้ลักษณะที่สำคัญและการกำจัดลักษณะที่ไม่ตรงประเด็นและลักษณะซ้ำซ้อนออกเพื่อช่วยลดขนาดเนื้อที่ในการค้นหาและไม่ต้องเสียเวลามากกับอัลกอริทึมวิเคราะห์ข้อมูล (Induction Algorithm) รวมทั้งเพิ่มความถูกต้องของผลลัพธ์ในกระบวนการวิเคราะห์ข้อมูล ซึ่งโดยทั่วไปแล้ววิธีการเลือกลักษณะแบ่งได้ 2 วิธี คือ วิธีการกรอง (Filter Approach) และวิธีการห่อหุ้ม (Wrapper Approach) [3, 10] วิธีการกรองจะพิจารณาเซตของข้อมูลโดยเป็นอิสระจากอัลกอริทึมวิเคราะห์ข้อมูล กล่าวคือจะทำงานจบในขั้นตอนการเตรียมข้อมูล (Preprocessing Step) ไม่อาศัยอัลกอริทึมวิเคราะห์ข้อมูลซึ่งเป็นการผลักภาระให้อัลกอริทึมวิเคราะห์ข้อมูลทำการตรวจสอบผลลัพธ์เพียงผู้เดียว ส่วนวิธีการห่อหุ้มนั้นต่างจากวิธีการกรอง คือ อัลกอริทึมวิเคราะห์ข้อมูลเป็นส่วนหนึ่งของการกระบวนการเลือกลักษณะ โดยอัลกอริทึมวิเคราะห์ข้อมูลจะทำหน้าที่เป็นตัวประเมินความถูกต้องและความตรงประเด็นของลักษณะ ด้วยเหตุนี้จึงทำให้วิธีการห่อหุ้มต้องเสียเวลาในการคำนวณมากเพราะต้องเรียกอัลกอริทึมวิเคราะห์ข้อมูลตลอดเวลา จึงทำให้วิธีการห่อหุ้มไม่เหมาะสมกับการทำงานในระบบข้อมูลที่มีขนาดใหญ่ และอาจจะเกิดความเอนเอียงได้เนื่องจากวิธีการเลือกลักษณะกับอัลกอริทึมวิเคราะห์ข้อมูลนั้นต้องมีความสอดคล้องกัน สำหรับวิธีการกรองแม้ว่าจะไม่มีการพิจารณาความถูกต้องของผลลัพธ์ แต่โดยส่วนใหญ่ก็นิยมใช้วิธีการกรองเพราะวิธีนี้ง่ายต่อการนำไปใช้งานและสามารถทำงานได้จริงในระบบที่มีขนาดใหญ่ รวมทั้งถ้าการประเมินลักษณะของวิธีการกรองนั้นเป็นไปอย่างมีเหตุมีผลก็น่าจะได้ผลลัพธ์ที่ดีด้วย [6]

ทฤษฎีรัฟเซต (Rough Set Theory) นำเสนอโดย Pawlak ในปี 1980 ซึ่งทฤษฎีรัฟเซตได้เตรียมเครื่องมือทางคณิตศาสตร์ไว้สำหรับการเลือกลักษณะที่เหมาะสม ซึ่งการเลือกลักษณะบนรัฟเซตเป็นงานที่เรียกว่าการค้นหาแอททริบิวต์รีดัก ดังนั้นการค้นหาแอททริบิวต์รีดักจึงเหมือนกับการเลือกลักษณะ ในสังคมรัฟเซตอัลกอริทึมเลือกลักษณะส่วนใหญ่เพื่อคำนวณหาแอทริบิวต์รีดักแบ่งออกได้ 2 วิธี คือ วิธีฟังก์ชันมองเห็นได้ (Discernibility Function Based Approach) และวิธีพื้นที่ทางบวก (Positive Region Based Approach) อย่างไรก็ตามทั้ง 2 วิธีคำนวณยุ่งยากและเสียเวลา

จำนวนมาก [4] ดังนั้น จึงมีงานวิจัยมาช่วยปรับปรุงวิธีการคำนวณหาแอททริบิวต์รีดักมากมาย เช่น การสร้างฟังก์ชันฮิวริสติกให้กับฟังก์ชันมองเห็นได้ [5, 7, 8] การใช้รีเลชันนอลแอลจีบรา โอเปอร์เรชัน (Relational Algebra Operation) เพื่อกำหนดนิยามแอททริบิวต์รีดักบนทฤษฎีรัฟเซต ขึ้นมาใหม่ [11] และการเสนอนิยามใหม่ของการขึ้นต่อกันระหว่างแอททริบิวต์ที่สัมพันธ์กัน (Relative Attribute Dependency) [4] เพื่อช่วยในการตรวจสอบเซตแอททริบิวต์รีดัก ซึ่งการแก้ปัญหาทั้งหมดมีทั้งข้อดีและข้อเสีย แต่ยังไม่มีงานวิจัยที่นำทฤษฎีฐานข้อมูลมาช่วยในการค้นหาแอททริบิวต์รีดักซึ่งเราเห็นว่านิยามของแอททริบิวต์รีดักบนทฤษฎีรัฟเซตสามารถเทียบเท่ากับทฤษฎีฐานข้อมูลเชิงสัมพันธ์ (Relational Database Theory) ดังนั้น งานวิจัยนี้จึงเสนอนิยามแอททริบิวต์รีดักขึ้นมาใหม่โดยอาศัยทฤษฎีฐานข้อมูลเชิงสัมพันธ์เรื่องฟังก์ชันการขึ้นต่อกัน (Functional Dependency) ซึ่งนิยามใหม่นี้นำไปสู่วิธีการในการคำนวณหาแอททริบิวต์รีดักที่มีประสิทธิภาพและได้ผลลัพธ์ครบถ้วนและถูกต้อง

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

งานวิจัยนี้เป็นการศึกษานิยามของแอททริบิวต์รีดักบนรัฟเซตเพื่อสร้างนิยามใหม่ของแอททริบิวต์รีดักโดยอาศัยทฤษฎีฟังก์ชันการขึ้นต่อกัน พร้อมทั้งศึกษาคุณสมบัติของฟังก์ชันการขึ้นต่อกันเพื่อนำไปใช้ในการสร้างวิธีการคำนวณหาแอททริบิวต์รีดักที่มีประสิทธิภาพ

1.3 สมมติฐานของการศึกษา

นิยามแอททริบิวต์รีดักที่กำหนดขึ้นเทียบเท่ากับนิยามแอททริบิวต์รีดักบนรัฟเซต และสามารถใช้นิยามนี้สร้างอัลกอริทึมค้นหาแอททริบิวต์รีดักที่มีประสิทธิภาพ โดยอัลกอริทึมจะให้ผลลัพธ์ครบถ้วนและถูกต้อง

1.4 ขอบเขตการศึกษา

งานวิจัยนี้เป็นการเชื่อมโยงนิยามแอททริบิวต์รีดักบนรัฟเซต โดยสร้างนิยามแอททริบิวต์รีดักบนทฤษฎีฟังก์ชันการขึ้นต่อกันที่เทียบเท่ากับนิยามแอททริบิวต์รีดักบนทฤษฎีรัฟเซต จากนั้นจะทำการเลือกทฤษฎีและอัลกอริทึมจากทฤษฎีฟังก์ชันการขึ้นต่อกันเพื่อใช้ในการหาแอททริบิวต์รีดักตามนิยามใหม่นี้

1.5 ส่วนประกอบของวิทยานิพนธ์

เนื้อหาส่วนที่เหลือของวิทยานิพนธ์ฉบับนี้ แบ่งตามบท ดังนี้

บทที่ 2 จะกล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้อง ซึ่งจะแบ่งเนื้อหาออกเป็น 3 ส่วน ดังนี้ ส่วนแรกจะกล่าวถึงการเลือกลักษณะ ส่วนที่ 2 จะกล่าวถึงทฤษฎีรัฟเซตซึ่งประกอบด้วยนิยามต่าง ๆ และคุณสมบัติของทฤษฎีรัฟเซต ส่วนสุดท้ายจะกล่าวถึงงานวิจัยที่เกี่ยวข้องกับการค้นหาแอททริบิวต์รีดักบนทฤษฎีรัฟเซต

บทที่ 3 ได้นำเสนอนิยามแอททริบิวต์รีดักบนฟังก์ชันการขึ้นต่อกัน โดยแบ่งเนื้อหาออกเป็น 4 ส่วน ดังนี้ ส่วนแรกจะกล่าวถึงทฤษฎีฟังก์ชันการขึ้นต่อกัน ส่วนที่ 2 จะกล่าวถึงคุณสมบัติของฟังก์ชันการขึ้นต่อกัน ส่วนที่ 3 จะกล่าวถึงความสอดคล้องกันระหว่างนิยามแอททริบิวต์รีดักบนทฤษฎีรัฟเซตและทฤษฎีฟังก์ชันการขึ้นต่อกัน ส่วนที่ 4 จะกล่าวถึงนิยามแอททริบิวต์รีดักบนฟังก์ชันการขึ้นต่อกัน และตัวอย่างแอททริบิวต์รีดักตามนิยามแอททริบิวต์รีดักบนฟังก์ชันการขึ้นต่อกัน

บทที่ 4 ได้นำเสนออัลกอริทึมค้นหาแอททริบิวต์รีดักบนทฤษฎีฟังก์ชันการขึ้นต่อกัน โดยแบ่งเนื้อหาออกเป็น 3 ส่วน ดังนี้ ส่วนแรกกล่าวถึงอัลกอริทึมค้นหาแอททริบิวต์รีดักบนทฤษฎีฟังก์ชันการขึ้นต่อกัน ส่วนที่ 2 จะกล่าวถึงตัวอย่างการทำงานของอัลกอริทึมค้นหาแอททริบิวต์รีดักบนทฤษฎีฟังก์ชันการขึ้นต่อกัน และส่วนสุดท้ายวิเคราะห์การทำงานของอัลกอริทึมค้นหาแอททริบิวต์รีดักบนทฤษฎีฟังก์ชันการขึ้นต่อกัน

บทที่ 5 จะทำการสรุปผลการวิจัย โดยได้แบ่งเนื้อหาออกเป็น 2 ส่วน ดังนี้ ส่วนแรกกล่าวถึงสรุปผลการวิจัย และส่วนที่ 2 จะกล่าวถึงข้อเสนอแนะสำหรับการทำวิจัย

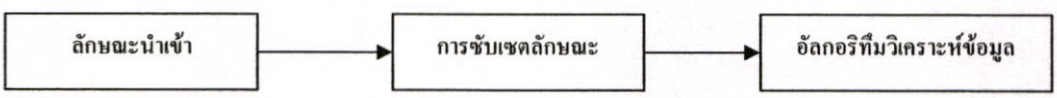
บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทที่ 2 นี้จะแบ่งเนื้อหาออกเป็น 3 ส่วน ดังนี้ ส่วนแรกจะกล่าวถึงการเลือกลักษณะ ส่วนที่ 2 จะกล่าวถึงทฤษฎีรัฟเซตซึ่งประกอบด้วยนิยามต่าง ๆ และคุณสมบัติของทฤษฎีรัฟเซต และส่วนสุดท้ายจะกล่าวถึงงานวิจัยที่เกี่ยวข้องกับวิธีการค้นหาแอททริบิวต์รีดักบนทฤษฎีรัฟเซต

2.1 การเลือกลักษณะ

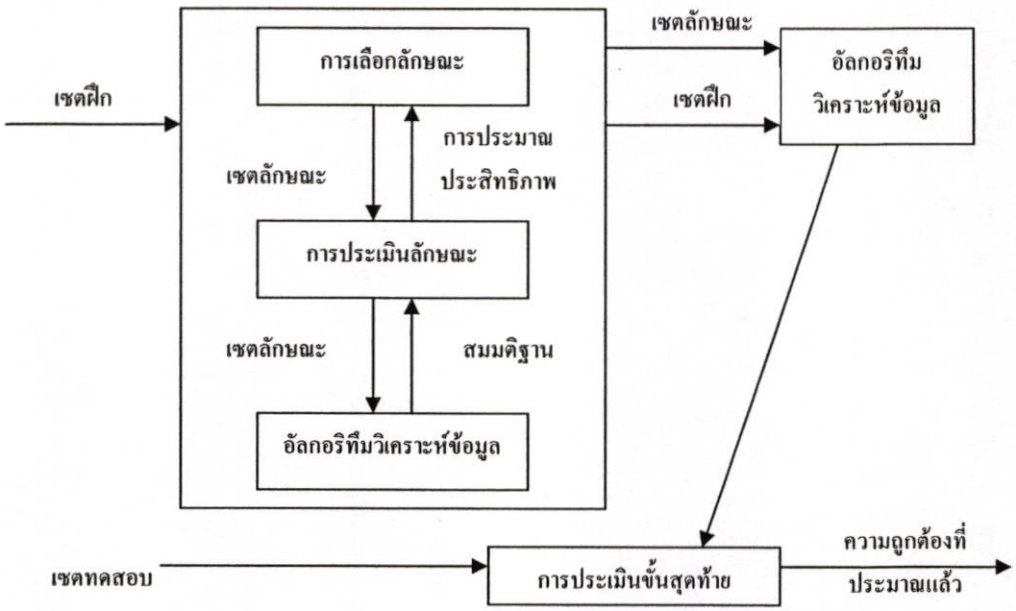
การเลือกลักษณะเป็นขั้นตอนหนึ่งของกระบวนการทำเหมืองข้อมูลเพื่อค้นหาเซตย่อยของลักษณะที่เหมาะสมที่สุด [3] ซึ่งการทำเหมืองข้อมูลเกี่ยวข้องกับการวิเคราะห์ข้อมูลจำนวนมหาศาลและมีความซับซ้อนมาก จึงกลายเป็นงานที่มีความสำคัญมากต่อการเพิ่มศักยภาพและความมีประสิทธิภาพของข้อมูล โดยส่วนใหญ่แล้วกระบวนการทำเหมืองข้อมูลนั้นประกอบด้วยลักษณะจำนวนมากรวมทั้งลักษณะที่ไม่ตรงประเด็น (Irrelevant Feature) และลักษณะที่ซ้ำซ้อน (Redundant Feature) ดังนั้นการเลือกลักษณะจึงเป็นงานที่สำคัญสำหรับการทำเหมืองข้อมูลเพื่อกำจัดลักษณะที่ไม่ตรงประเด็นและลดลักษณะที่ซ้ำซ้อน ซึ่งจะช่วยปรับปรุงศักยภาพของการทำเหมืองข้อมูลโดยการลดขนาดเนื้อที่และเวลาที่ใช้ในการวิเคราะห์ข้อมูล ซึ่งถ้าหากการเลือกลักษณะทำงานได้อย่างมีประสิทธิภาพก็จะทำให้อัลกอริทึมวิเคราะห์ข้อมูลสามารถทำงานบนข้อมูลที่มีลักษณะที่ตรงประเด็นและมีความถูกต้องสูง นอกจากนี้จะทำให้ได้ตัวแทนที่มีศักยภาพและง่ายต่อความเข้าใจ [6, 10] วิธีการเลือกลักษณะแบ่งออกได้ 2 วิธี ดังนี้ [3, 10]

1. วิธีการกรอง จะพิจารณาเซตของข้อมูลโดยเป็นอิสระจากอัลกอริทึมวิเคราะห์ข้อมูล กล่าวคือ จะทำงานจบในขั้นตอนการเตรียมข้อมูล โดยไม่มีการอาศัยอัลกอริทึมวิเคราะห์ข้อมูล ซึ่งเป็นการผลักภาระให้อัลกอริทึมวิเคราะห์ข้อมูล การเลือกลักษณะโดยวิธีการกรอง แสดงดังรูปที่ 2.1



รูปที่ 2.1 การเลือกลักษณะของวิธีการกรอง

2. วิธีการห่อหุ้ม จะนำอัลกอริทึมวิเคราะห์ข้อมูลมาเป็นส่วนหนึ่งของกระบวนการเลือกลักษณะ โดยอัลกอริทึมวิเคราะห์ข้อมูลจะทำหน้าที่เป็นตัวประเมินความถูกต้องและความตรงประเด็นของลักษณะ การเลือกลักษณะโดยวิธีการห่อหุ้ม แสดงดังรูปที่ 2.2



รูปที่ 2.2 การเลือกลักษณะของวิธีการห่อหุ้ม

ด้วยเหตุนี้ทำให้การเลือกลักษณะโดยวิธีการห่อหุ้มต้องเสียเวลาในการคำนวณมาก เพราะต้องเรียกอัลกอริทึมวิเคราะห์ข้อมูลตลอดเวลา จึงทำให้วิธีการห่อหุ้มไม่เหมาะสมกับการทำงานในระบบข้อมูลที่มีขนาดใหญ่ สำหรับการเลือกลักษณะโดยวิธีการกรองแม้ว่าจะไม่มีการพิจารณาความถูกต้องของผลลัพธ์ แต่โดยส่วนใหญ่ก็นิยมใช้วิธีการกรองเพราะวิธีนี้ง่ายต่อการนำไปใช้งานและสามารถทำงานได้จริงในระบบที่มีขนาดใหญ่ รวมทั้งถ้าการประเมินลักษณะของวิธีการกรองนั้นเป็นไปอย่างมีเหตุมีผลก็น่าจะได้ผลลัพธ์ที่ดีด้วย [6]

2.2 ทฤษฎีรัฟเซต

ทฤษฎีรัฟเซตนำเสนอโดย Pawlak ในปี 1980 ซึ่งทฤษฎีรัฟเซตถูกนำไปใช้กับแอพพิเคชันต่าง ๆ เช่น การเรียนรู้ของเครื่องมือ การได้มาซึ่งองค์ความรู้ การวิเคราะห์การตัดสินใจ การค้นหาองค์ความรู้จากฐานข้อมูล และระบบผู้เชี่ยวชาญ เป็นต้น นิยามพื้นฐานของทฤษฎีรัฟเซต [4, 5, 7, 8, 11] มีรายละเอียด ดังนี้

นิยามที่ 2.1 ตารางตัดสินใจ (Decision Table)
กำหนดให้ ตารางตัดสินใจเป็นระบบสารสนเทศระบบหนึ่ง ซึ่งประกอบด้วยเซตของ U, A, V และ f เขียนแทนด้วย T (U, A = C $\cup$ D , V, f) ซึ่ง C $\cap$ D = ϕ โดยที่ U เป็นเซตจำกัดและไม่เท่ากับเซตว่าง ซึ่งเรียกว่า เอกภพ เขียนแทนด้วย U = $\{x_1, x_2, \ldots, x_i\}$ และสมาชิกของเอกภพจะเรียกว่า ออบเจค และ i เป็นจำนวนออบเจค, C เป็นเซตแอททริบิวต์เงื่อนไข ซึ่งเป็นเซตจำกัดและไม่เท่ากับเซตว่าง เขียนแทนด้วย C = $\{a_1, a_2, \ldots, a_n\}$ และ n เป็นจำนวนแอททริบิวต์เงื่อนไข, D เป็นแอททริบิวต์ตัดสินใจ ซึ่งเป็นเซตจำกัดและไม่เท่ากับเซตว่าง เขียนแทนด้วย D = $\{ d_1, d_2, \ldots, d_m \}$ และ m เป็นจำนวนแอททริบิวต์ตัดสินใจ, V เป็นค่าโดเมนของแต่ละแอททริบิวต์ A เขียนแทนด้วย $V = \bigcup_{p \in A} V_p$ ซึ่ง V_p เป็นค่าโดเมนของแอททริบิวต์ p และ f เป็นฟังก์ชันระหว่างเอกภพ U กับแอททริบิวต์ A สอดคล้องกับค่าโดเมนของแอททริบิวต์ V เขียนแทนด้วย $f : U \times A \rightarrow V$ ซึ่ง $f(x_i, q)$ เป็นฟังก์ชันระหว่างออบเจค i กับแอททริบิวต์ q สำหรับทุก ๆ $q \in A$ และ $x_i \in U$

ตัวอย่างที่ 2.1 ตารางตัดสินใจ

พิจารณาตารางตัดสินใจดังตารางที่ 2.1 กำหนดให้ แอททริบิวต์ Door, Size และ Cylinder เป็นแอททริบิวต์เงื่อนไข และแอททริบิวต์ Mileage เป็นแอททริบิวต์ตัดสินใจ ตามนิยามที่ 2.1 จะได้ว่า $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$

$A = C \cup D = \{a_1, a_2, a_3, a_4\} = \{\text{Door, Size, Cylinder, Mileage}\}$

$C = \{\text{Door, Size, Cylinder}\}$

$D = \{\text{Mileage}\}$

$V = \{V_{a_1}, V_{a_2}, V_{a_3}, V_{a_4}\}$

$V_{a_1} = V_{\text{Door}} = \{2,4\}$

$V_{a_2} = V_{\text{Size}} = \{\text{Compact, Sub}\}$

$V_{a_3} = V_{\text{Cylinder}} = \{4,6\}$

$V_{a_4} = V_{\text{Mileage}} = \{\text{Low, High}\}$

$f(x_i,q) \in V_q$ สมมติ $i = 1$ และ $q = a_1$ จะได้ $f(x_1, a_1) \in V_{a_1}$

$f(x_1, a_1) = f(x_1, \text{Door}) = 2$

ตารางที่ 2.1 แสดงตารางตัดสินใจของข้อมูลรถเพื่อทำนายระยะทาง ที่มีแอททริบิวต์ Door, Size และ Cylinder เป็นแอททริบิวต์เงื่อนไข และแอททริบิวต์ Mileage เป็นแอททริบิวต์ตัดสินใจ

U	Door	Size	Cylinder	Mileage
x_1	2	Compact	4	High
x_2	4	Sub	6	Low
x_3	4	Compact	4	High
x_4	2	Compact	6	Low
x_5	4	Compact	4	Low
x_6	4	Compact	4	High
x_7	4	Sub	6	Low
x_8	2	Sub	6	Low

นิยามที่ 2.2 ความสัมพันธ์ไม่สามารถมองเห็นได้ (Indiscernibility Relation)
กำหนดให้ P เป็นเซตย่อยของแอททริบิวต์ A เขียนแทนด้วย $P \subseteq A$ และออบเจค i และออบเจค j เป็นสมาชิกของเอกภพ เขียนแทนด้วย $x_i,\ x_j \in U$ จะกล่าวว่า ความสัมพันธ์ไม่สามารถมองเห็นได้ของเซตแอททริบิวต์ P เขียนแทนด้วย IND(P) เป็นเซตออบเจคของเอกภพ ซึ่งออบเจค 2 ออบเจคใด ๆ ที่มีค่าโดเมนของเซตแอททริบิวต์ P เหมือนกัน เขียนแทนด้วย

$$IND(P) = \{(x_i, x_j) : (x_i, x_j) \in U \times U, a \in P, f(x_i, a) = f(x_j, a)\}$$

และ U/IND(P) เป็นเซตคลาสสมมูลกันทั้งหมดของความสัมพันธ์ไม่สามารถมองเห็นได้

<u>ตัวอย่างที่ 2.2</u> แสดงความสัมพันธ์ที่ไม่สามารถมองเห็นได้จากข้อมูลในตารางที่ 2.1

จากตารางที่ 2.1 เมื่อพิจารณาความสัมพันธ์ไม่สามารถมองเห็นได้ตามนิยามที่ 2.2 จะได้ว่า

$U/IND(\{Door\}) = \{\{x_1, x_4, x_8\}, \{x_2, x_3, x_5, x_6, x_7\}\}$

$U/IND(\{Size\}) = \{\{x_1, x_3, x_4, x_5, x_6\}, \{x_2, x_7, x_8\}\}$

$U/IND(\{Size, Cylinder\}) = \{\{x_1, x_3, x_5, x_6\}, \{x_2, x_7, x_8\}, \{x_4\}\}$

$U/IND(\{Door, Size, Cylinder\}) = \{\{x_1\}, \{x_2, x_7\}, \{x_3, x_5, x_6\}, \{x_4\}, \{x_8\}\}$

นิยามที่ 2.3 ค่าประมาณขอบเขตล่าง (Lower Approximate)
กำหนดให้ R เป็นเซตย่อยของแอททริบิวต์ C เขียนแทนด้วย $R \subseteq C$ และ X เป็นเซตย่อยของเอกภพ เขียนแทนด้วย $X \subseteq U$ จะกล่าวว่า ค่าประมาณขอบเขตล่างของออบเจค X สอดคล้องกับเซตแอททริบิวต์ R เขียนแทนด้วย $\underline{RX}$ เป็นเซตออบเจคทั้งหมดของเอกภพที่สามารถแบ่งแยกคลาสตามค่าแอททริบิวต์ตัดสินใจได้อย่างแน่นอนสอดคล้องกับค่าเซตแอททริบิวต์ R เขียนแทนด้วย

$$\underline{RX} = \bigcup\{Y \in U / IND(R) : Y \subseteq X\}$$

นิยามที่ 2.4 พื้นที่ทางบวก (Positive Region)

พื้นที่ทางบวกของแอททริบิวต์ตัดสินใจ D สอดคล้องกับค่าเซตแอททริบิวต์ R เขียนแทนด้วย $POS_R(D)$ เป็นออบเจคทั้งหมดที่สามารถแบ่งแยกคลาสได้อย่างแน่นอนตามค่าโดเมนแอททริบิวต์ตัดสินใจที่เป็นไปได้ทั้งหมดสอดคล้องกับค่าเซตแอททริบิวต์ R เขียนแทนด้วย

$$POS_R(D) = \bigcup_{X \in U/IND(D)} \underline{RX}$$

<u>ตัวอย่างที่ 2.3</u> ค่าประมาณขอบเขตล่างและพื้นที่ทางบวก

จากตารางที่ 2.1 กำหนดให้ R = C = {Door, Size, Cylinder} และ D = {Mileage} ตามนิยามที่ 2.2, นิยามที่ 2.3 และนิยามที่ 2.4 จะได้ว่า

$U/IND(R) = \{\{x_1\}, \{x_2, x_7\}, \{x_3, x_5, x_6\}, \{x_4\}, \{x_8\}\}$

$U/IND(D) = \{\{x_1, x_3, x_6\}, \{x_2, x_4, x_5, x_7, x_8\}\}$

กำหนดให้ $X_1 = U/IND[Mileage = High] = \{x_1, x_3, x_6\}$

และ $X_2 = U/IND[Mileage = Low] = \{x_2, x_4, x_5, x_7, x_8\}$

$$\underline{RX_1} = \bigcup \{U/IND(R) : U/IND(R) \subseteq U/IND[Mileage = High]\}$$

$$\underline{RX_1} = (\{x_1\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_2, x_7\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_3, x_5, x_6\} \subseteq \{x_1, x_3, x_6\})$$
$$\cup (\{x_4\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_8\} \subseteq \{x_1, x_3, x_6\})$$

จะได้ว่า $\underline{RX_1} = \{x_1\}$

$$\underline{RX_2} = \bigcup \{U/IND(R) : U/IND(R) \subseteq U/IND[Mileage = Low]\}$$

$$\underline{RX_2} = (\{x_1\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_2, x_7\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$
$$\cup (\{x_3, x_5, x_6\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_4\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$
$$\cup (\{x_8\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$

จะได้ว่า $\underline{RX_2} = \{x_2, x_7, x_4, x_8\}$

ดังนั้น $POS_R(D) = \bigcup_{X \in U/IND(D)} \underline{RX} = \underline{RX_1} \cup \underline{RX_2} = \{x_1, x_2, x_4, x_7, x_8\}$

จากตัวอย่างแสดงให้เห็นว่า $POS_R(D) = \{x_1, x_2, x_4, x_7, x_8\}$ แสดงว่าข้อมูลที่ได้รับจากเซตแอททริบิวต์เงื่อนไขที่ประกอบด้วยแอททริบิวต์ Door, Size และ Cylinder เป็นข้อมูลไม่สมบูรณ์ เนื่องจากสามารถบ่งบอกถึงข้อมูลรถได้เพียง 5 คัน นั่นคือ เซตแอททริบิวต์เงื่อนไขดังกล่าวจะดีพอสำหรับการสร้างโมเดลคลาสของข้อมูลรถ 5 คันเท่านั้น ดังนั้น เพื่อที่จะแบ่งแยกคลาสตัดสินใจของข้อมูลรถ x_3, x_5, x_6 และสามารถบ่งบอกข้อมูลรถได้ทั้งหมด 8 คันแสดงว่าจะต้องเก็บรวบรวมข้อมูลรถเพิ่มเติม สมมติว่าทำการเก็บข้อมูลน้ำหนักรถเพิ่มให้กับรถแต่ละคัน แสดงดังตารางที่ 2.2

ตารางที่ 2.2 แสดงตารางตัดสินใจของข้อมูลรถเพื่อทำนายระยะทาง ที่มีแอททริบิวต์ Weight, Door, Size และ Cylinder เป็นแอททริบิวต์เงื่อนไข และแอททริบิวต์ Mileage เป็นแอททริบิวต์ตัดสินใจ

U	Weight	Door	Size	Cylinder	Mileage
x_1	Low	2	Compact	4	High
x_2	Low	4	Sub	6	Low
x_3	Med	4	Compact	4	High
x_4	High	2	Compact	6	Low
x_5	High	4	Compact	4	Low
x_6	Low	4	Compact	4	High
x_7	High	4	Sub	6	Low
x_8	Low	2	Sub	6	Low

จากตารางที่ 2.2 กำหนดให้ $R = C = \{\text{Weight, Door, Size, Cylinder}\}$ และ $D = \{\text{Mileage}\}$

$U/IND(R) = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}, \{x_7\}, \{x_8\}\}$

$U/IND(D) = \{\{x_1, x_3, x_6\}, \{x_2, x_4, x_5, x_7, x_8\}\}$

กำหนดให้ $X_1 = U/IND[\text{Mileage} = \text{High}] = \{x_1, x_3, x_6\}$

และ $X_2 = U/IND[\text{Mileage} = \text{Low}] = \{x_2, x_4, x_5, x_7, x_8\}$

$$\underline{RX_1} = \bigcup \{U/IND(R) : U/IND(R) \subseteq U/IND[\text{Mileage} = \text{High}]\}$$

$$\underline{RX_1} = (\{x_1\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_2\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_3\} \subseteq \{x_1, x_3, x_6\})$$
$$\cup (\{x_4\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_5\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_6\} \subseteq \{x_1, x_3, x_6\})$$
$$\cup (\{x_7\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_8\} \subseteq \{x_1, x_3, x_6\})$$

จะได้ว่า $\underline{RX_1} = \{x_1, x_3, x_6\}$

$$\underline{RX_2} = \bigcup \{U/IND(R) : U/IND(R) \subseteq U/IND[\text{Mileage} = \text{Low}]\}$$

$$\underline{RX_2} = (\{x_1\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_2\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$
$$\cup (\{x_3\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_4\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$
$$\cup (\{x_5\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_6\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$
$$\cup (\{x_7\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_8\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$

จะได้ว่า $\underline{RX_2} = \{x_2, x_4, x_5, x_7, x_8\}$

ดังนั้น $POS_R(D) = \bigcup_{X \in U/IND(D)} \underline{RX} = \underline{RX_1} \cup \underline{RX_2} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$

หลังจากเพิ่มแอททริบิวต์น้ำหนักให้กับข้อมูลรถแต่ละคันแล้วทำให้สามารถบ่งบอกข้อมูลรถได้ทั้งหมด 8 คัน ดังนั้น จะสามารถสร้างโมเดลคลาสของข้อมูลรถทั้งหมด 8 คันได้ ซึ่งกล่าวได้ว่าเป็นคุณสมบัติที่ดีข้อหนึ่งของทฤษฎีรัฟเซต คือ สามารถบอกได้ว่าข้อมูลสมบูรณ์พอที่จะสร้างโมเดลคลาสหรือไม่ กล่าวคือ ถ้าข้อมูลไม่สมบูรณ์เราจะต้องเก็บรวบรวมข้อมูลมากขึ้นเพื่อที่จะสร้างโมเดลคลาสที่ดี หรือกล่าวอีกนัยนึงว่า ถ้าข้อมูลสมบูรณ์ ทฤษฎีรัฟเซตสามารถบอกได้ว่าข้อมูลมีความซ้ำซ้อนหรือเพียงพอที่จะค้นหาข้อมูลที่มีความจำเป็นและเป็นข้อมูลที่น้อยที่สุดสำหรับสร้างโมเดลคลาสหรือไม่ ซึ่งคุณสมบัติข้อนี้มีความสำคัญมากกับแอพพิเคชัน ขณะที่โดเมนองค์ความรู้ถูกจำกัดเป็นอย่างมากและงานเก็บรวบรวมข้อมูลเป็นงานหนักรวมทั้งเสียค่าใช้จ่ายแพง ดังนั้นคุณสมบัติของทฤษฎีรัฟเซตจะทำให้เราแน่ใจว่าข้อมูลที่เก็บรวบรวมดีเพียงพอที่จะสร้างโมเดลคลาสโดยปราศจากการทำลายโมเดลคลาสที่ถูกต้องและไม่ทำให้เสียเวลาจำนวนมาก รวมทั้งความพยายามที่จะเก็บข้อมูลพิเศษเพื่อนำมาสร้างโมเดลคลาสให้สมบูรณ์ ดังนั้น แสดงให้เห็นว่าข้อมูลในตารางเป็นข้อมูลที่สมบูรณ์และเหมาะสมกับการนำมาสร้างคลาสตัดสินใจที่ดีเพื่อทำให้ได้รับแอททริบิวต์รีดักที่เหมาะสมที่สุด

นอกจากนี้ทฤษฎีรัฟเซตยังสามารถแบ่งแยกความสำคัญของลักษณะหนึ่งออกเป็น 2 ประเภท คือ ลักษณะตรงประเด็น (Relevant Feature) และลักษณะไม่ตรงประเด็น (Irrelevant Feature) ซึ่งลักษณะไม่ตรงประเด็นเรียกว่าแอททริบิวต์ไม่จำเป็น (Dispensable attribute) ซึ่งแอททริบิวต์ไม่จำเป็นนี้ทำให้เกิดข้อมูลที่ซ้ำซ้อนและทำลายความถูกต้องของโมเดลคลาส ดังนั้นควรจะกำจัดแอททริบิวต์ไม่จำเป็นทิ้ง ลักษณะตรงประเด็นนั้นจะแบ่งได้ 2 แบบ คือ 1. ลักษณะแข็งแรง (Strong Feature) และลักษณะอ่อนแอ (Weak Feature) ลักษณะแข็งแรงหมายถึงลักษณะหลักหรือเรียกว่า แอททริบิวต์คอร์ (Core Attribute) ซึ่งแอททริบิวต์คอร์นี้จะประกอบด้วยข้อมูลที่มีความสำคัญในการสร้างโมเดลคลาสอย่างความถูกต้อง ดังนั้น แอททริบิวต์คอร์นี้ควรจะเก็บไว้อย่างแน่นอน และ 2. ลักษณะอ่อนแอเป็นลักษณะที่อยู่ระหว่างลักษณะแข็งแรงและลักษณะไม่จำเป็นซึ่งเรียกว่าแอททริบิวต์รีดัก (Reduct Attribute) ซึ่งแอททริบิวต์รีดักนี้ขึ้นอยู่กับว่านำแอททริบิวต์ใดมาประกอบร่วมกันบ้างเพื่อให้เกิดการสร้างโมเดลคลาสสมบูรณ์ ซึ่งบางครั้งเซตแอทริบิวต์รีดักหนึ่งมีความจำเป็นกับสถานการณ์หนึ่งขณะที่อีกสถานการณ์หนึ่งไม่มีความจำเป็นซึ่งสถานการณ์ต่างๆนั้นขึ้นอยู่กับโดเมนปัญหาที่สนใจและบางสถานการณ์อาจมีเซตแอททริบิวต์รีดักได้มากกว่า 1 เซตแอททริบิวต์

รวมทั้งจะกล่าวได้ว่าคำตอบแอททริบิวต์รีดักที่เป็นไปได้ของทฤษฎีรัฟเซตสามารถแบ่งออกเป็น 3 ประเภท ดังนี้

1. แอททริบิวต์รีดัก = แอททริบิวต์คอร์ทั้งหมด
2. แอททริบิวต์รีดัก = แอททริบิวต์คอร์บวกร่วมกับแอททริบิวต์ที่มีลักษณะอ่อนแอ
3. แอททริบิวต์รีดัก = ลักษณะอ่อนแอทั้งหมด

ดังนั้นในการค้นหาแอททริบิวต์รีดักโดยส่วนใหญ่แล้วจะเริ่มทำการค้นหาแอททริบิวต์คอร์ก่อนเป็นอันดับแรกเพื่อช่วยลดทางเลือกของคำตอบแอททริบิวต์รีดัก

นิยามที่ 2.5 แอททริบิวต์คอร์

แอททริบิวต์ $C_j \in C$ เป็นแอททริบิวต์คอร์ ถ้า $POS_C(D) \neq POS_{C-C_j}(D)$

นิยามที่ 2.6 แอททริบิวต์ไม่จำเป็น

แอททริบิวต์ $C_j \in C$ เป็นแอททริบิวต์ไม่จำเป็น ถ้า $POS_C(D) = POS_{C-C_j}(D)$

นิยามที่ 2.7 แอททริบิวต์รีดัก

แอททริบิวต์ $C_j \in C$ เป็นแอททริบิวต์รีดัก ถ้า C_j เป็นส่วนหนึ่งของแอททริบิวต์รีดัก

<u>ตัวอย่างที่ 2.4</u> แอททริบิวต์คอร์และแอททริบิวต์ไม่จำเป็นจากข้อมูลในตารางที่ 2.2

จากตารางที่ 2.2 จะได้ว่า C = {Weight, Door, Size, Cylinder} และ D = {Mileage}
พิจารณาแอททริบิวต์คอร์และแอททริบิวต์ไม่จำเป็น ตามนิยามที่ 2.5 และนิยามที่ 2.6 จะได้ว่า

$U/IND(C) = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}, \{x_7\}, \{x_8\}\}$

$U/IND(D) = \{\{x_1, x_3, x_6\}, \{x_2, x_4, x_5, x_7, x_8\}\}$

กำหนดให้ $X_1 = U/IND[Mileage = High] = \{x_1, x_3, x_6\}$

และ $X_2 = U/IND[Mileage = Low] = \{x_2, x_4, x_5, x_7, x_8\}$

$$\underline{CX_1} = \bigcup \{U/IND(C) : U/IND(C) \subseteq U/IND[Mileage = High]\}$$

$$\underline{CX_1} = (\{x_1\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_2\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_3\} \subseteq \{x_1, x_3, x_6\})$$
$$\cup (\{x_4\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_5\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_6\} \subseteq \{x_1, x_3, x_6\})$$
$$\cup (\{x_7\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_8\} \subseteq \{x_1, x_3, x_6\})$$

จะได้ว่า $\underline{CX_1} = \{x_1, x_3, x_6\}$

$$\underline{CX_2} = \bigcup \{U/IND(C) : U/IND(C) \subseteq U/IND[Mileage = Low]\}$$

$$\underline{CX_2} = (\{x_1\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_2\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$
$$\cup (\{x_3\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_4\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$
$$\cup (\{x_5\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_6\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$
$$\cup (\{x_7\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_8\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$

จะได้ว่า $\underline{CX_2} = \{x_2, x_4, x_5, x_7, x_8\}$

$$POS_C(D) = \bigcup_{X \in U/IND(D)} \underline{CX} = \underline{CX_1} \cup \underline{CX_2} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$$

ดังนั้น $POS_C(D) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$

พิจารณา $C_1 = \{Weight\}$ ว่าเป็นแอททริบิวต์คอร์หรือไม่

$POS_{\{C\text{-}Weight\}}(D) = POS_{\{Door, Size, Cylinder\}}(D)$ สมมติให้ $R_1 = \{Door, Size, Cylinder\}$

$U/IND(R_1) = \{\{x_1\}, \{x_2, x_7\}, \{x_3, x_5, x_6\}, \{x_4\}, \{x_8\}\}$ แสดงดังตารางที่ 2.1

$$\underline{R_1X_1} = \bigcup \{U/IND(R_1) : U/IND(R_1) \subseteq U/IND[Mileage = High]\}$$

$$\underline{R_1X_1} = (\{x_1\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_2, x_7\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_3, x_5, x_6\} \subseteq \{x_1, x_3, x_6\})$$
$$\cup (\{x_4\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_8\} \subseteq \{x_1, x_3, x_6\})$$

จะได้ว่า $\underline{R_1X_1} = \{x_1\}$

$$\underline{R_1X_2} = \bigcup \{U/IND(R_1) : U/IND(R_1) \subseteq U/IND[Mileage = Low]\}$$

$$\underline{R_1X_2} = (\{x_1\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_2, x_7\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$
$$\cup (\{x_3, x_5, x_6\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_4\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$
$$\cup (\{x_8\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$

จะได้ว่า $\underline{R_1X_2} = \{x_2, x_7, x_4, x_8\}$

$$POS_{R_1}(D) = \bigcup_{X \in U/IND(D)} \underline{R_1X} = \underline{R_1X_1} \cup \underline{R_1X_2} = \{x_1, x_2, x_4, x_7, x_8\}$$

ดังนั้น $POS_C(D) \neq POS_{\{C-Weight\}}(D)$ แสดงว่าแอททริบิวต์ Weight เป็นแอททริบิวต์คอร์

พิจารณา $C_2 = \{Door\}$

$POS_{\{C\text{-}Door\}}(D) = POS_{\{Weight, Size, Cylinder\}}(D)$ สมมติให้ $R_2 = \{Weight, Size, Cylinder\}$

$U/IND(R_2) = \{\{x_1, x_6\}, \{x_2, x_8\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_7\}\}$ แสดงดังตารางที่ 2.3

$$\underline{R_2X_1} = \bigcup \{U/IND(R_2) : U/IND(R_2) \subseteq U/IND[Mileage = High]\}$$

$$\underline{R_2X_1} = (\{x_1, x_6\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_2, x_8\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_3\} \subseteq \{x_1, x_3, x_6\})$$
$$\cup (\{x_4\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_5\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_7\} \subseteq \{x_1, x_3, x_6\})$$

จะได้ว่า $\underline{R_2X_1} = \{x_1, x_6, x_3\}$

$\underline{R_2X_2} = \bigcup\{U/IND(R_2) : U/IND(R_2) \subseteq U/IND[Mileage = Low]\}$

$\underline{R_2X_2} = (\{x_1, x_6\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_2, x_8\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$
$\cup (\{x_3\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_4\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$
$\cup (\{x_5\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_7\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$

จะได้ว่า $\underline{R_2X_2} = \{x_2, x_8, x_4, x_5, x_7\}$

$POS_{R_2}(D) = \bigcup_{X \in U/IND(D)} \underline{R_2X} = \underline{R_2X_1} \cup \underline{R_2X_2} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$

ดังนั้น $POS_C(D) = POS_{\{C-Door\}}(D)$ แสดงว่าแอททริบิวต์ Door เป็นแอททริบิวต์ไม่จำเป็น

ตารางที่ 2.3 แสดงตารางตัดสินใจของข้อมูลรถเพื่อทำนายระยะทาง ที่มีแอททริบิวต์ Weight, Size และ Cylinder เป็นแอททริบิวต์เงื่อนไข และแอททริบิวต์ Mileage เป็นแอททริบิวต์ตัดสินใจ

U	Weight	Size	Cylinder	Mileage
x_1	Low	Compact	4	High
x_2	Low	Sub	6	Low
x_3	Med	Compact	4	High
x_4	High	Compact	6	Low
x_5	High	Compact	4	Low
x_6	Low	Compact	4	High
x_7	High	Sub	6	Low
x_8	Low	Sub	6	Low

พิจารณา C_3 = {Size}

$POS_{\{C-Size\}}(D) = POS_{\{Weight, Door, Cylinder\}}(D)$ สมมติให้ R_3 = {Weight, Door, Cylinder}

$U/IND(R_3)$ = $\{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}, \{x_7\}, \{x_8\}\}$ แสดงดังตารางที่ 2.4

$\underline{R_3X_1} = \bigcup\{U/IND(R_3) : U/IND(R_3) \subseteq U/IND[Mileage = High]\}$

$\underline{R_3X_1} = (\{x_1\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_2\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_3\} \subseteq \{x_1, x_3, x_6\})$
$\cup (\{x_4\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_5\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_6\} \subseteq \{x_1, x_3, x_6\})$
$\cup (\{x_7\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_8\} \subseteq \{x_1, x_3, x_6\})$

$\underline{R_3X_1} = \{x_1, x_3, x_6\}$

$\underline{R_3X_2} = \bigcup\{U/IND(R_3) : U/IND(R_3) \subseteq U/IND[Mileage = Low]\}$

$\underline{R_3X_2} = (\{x_1\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_2\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$
$\cup (\{x_3\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_4\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$
$\cup (\{x_5\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_6\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$
$\cup (\{x_7\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_8\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$

จะได้ว่า $\underline{R_3X_2} = \{x_2, x_4, x_5, x_7, x_8\}$

$POS_{R_3}(D) = \bigcup_{X \in U/IND(D)} \underline{R_3X} = \underline{R_3X_1} \cup \underline{R_3X_2} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$

ดังนั้น $POS_C(D) = POS_{\{C-Size\}}(D)$ แสดงว่าแอททริบิวต์ Size เป็นแอททริบิวต์ไม่จำเป็น

ตารางที่ 2.4 แสดงตารางตัดสินใจของข้อมูลรถเพื่อทำนายระยะทาง ที่มีแอททริบิวต์ Weight, Door และ Cylinder เป็นแอททริบิวต์เงื่อนไข และแอททริบิวต์ Mileage เป็นแอททริบิวต์ตัดสินใจ

U	Weight	Door	Cylinder	Mileage
x_1	Low	2	4	High
x_2	Low	4	6	Low
x_3	Med	4	4	High
x_4	High	2	6	Low
x_5	High	4	4	Low
x_6	Low	4	4	High
x_7	High	4	6	Low
x_8	Low	2	6	Low

พิจารณา C_4 = {Cylinder}

$POS_{\{C-\ Cylinder\}}(D) = POS_{\{Weight,\ Door,\ Size\}}(D)$ สมมติให้ R_4 = {Weight, Door, Size}

$U/IND(R_4) = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}, \{x_7\}, \{x_8\}\}$ แสดงดังตารางที่ 2.5

$\underline{R_4X_1} = \bigcup\{U/IND(R_4) : U/IND(R_4) \subseteq U/IND[Mileage = High]\}$

$\underline{R_4X_1} = (\{x_1\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_2\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_3\} \subseteq \{x_1, x_3, x_6\})$
$\cup (\{x_4\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_5\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_6\} \subseteq \{x_1, x_3, x_6\})$
$\cup (\{x_7\} \subseteq \{x_1, x_3, x_6\}) \cup (\{x_8\} \subseteq \{x_1, x_3, x_6\})$

จะได้ว่า $\underline{R_4X_1} = \{x_1, x_3, x_6\}$

$$\underline{R_4X_2} = \bigcup\{U/IND(R_4) : U/IND(R_4) \subseteq U/IND[Mileage = Low]\}$$

$$\underline{R_4X_2} = (\{x_1\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_2\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$

$$\cup(\{x_3\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_4\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$

$$\cup(\{x_5\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_6\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$

$$\cup(\{x_7\} \subseteq \{x_2, x_4, x_5, x_7, x_8\}) \cup (\{x_8\} \subseteq \{x_2, x_4, x_5, x_7, x_8\})$$

$$\underline{R_4X_2} = \{x_2, x_4, x_5, x_7, x_8\}$$

$$POS_{R_4}(D) = \bigcup_{X \in U/IND(D)} \underline{R_4X} = \underline{R_4X_1} \cup \underline{R_4X_2} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$$

ดังนั้น $POS_C(D) = POS_{\{C-Cylinder\}}(D)$ แสดงว่าแอททริบิวต์ Cylinder เป็นแอททริบิวต์ไม่จำเป็น

จากตัวอย่างสรุปได้ว่า แอททริบิวต์ Weight เป็นแอททริบิวต์คอร์ ซึ่งเป็นส่วนหนึ่งของแอททริบิวต์รีดัก

ตารางที่ 2.5 แสดงตารางตัดสินใจของข้อมูลรถเพื่อทำนายระยะทาง ที่มีแอททริบิวต์ Weight, Door และ Size เป็นแอททริบิวต์เงื่อนไข และแอททริบิวต์ Mileage เป็นแอททริบิวต์ตัดสินใจ

U	Weight	Door	Size	Mileage
x_1	Low	2	Compact	High
x_2	Low	4	Sub	Low
x_3	Med	4	Compact	High
x_4	High	2	Compact	Low
x_5	High	4	Compact	Low
x_6	Low	4	Compact	High
x_7	High	4	Sub	Low
x_8	Low	2	Sub	Low

จากทฤษฎีรัฟเซตข้างต้นจะทำให้สรุปคุณสมบัติที่ดีของทฤษฎีรัฟเซต ได้ 2 ข้อ ดังนี้

1. ทฤษฎีรัฟเซตทำให้แน่ใจว่าข้อมูลที่เก็บรวบรวมมาก่อนนำมาค้นหาแอททริบิวต์รีดักเพียงพอหรือไม่ กล่าวคือ ข้อมูลตั้งต้นทั้งหมดของตารางตัดสินใจก่อนนำมาค้นหาแอททริบิวต์รีดักนั้นต้องเป็นข้อมูลสอดคล้อง (Consistency) นั่นคือ พื้นที่ทางบวกของแอททริบิวต์ตัดสินใจ D ภายใต้แอททริบิวต์เงื่อนไข C ทั้งหมด ($POS_C(D)$) จะเท่ากับเซตของแถวข้อมูลทั้งหมดในตารางตัดสินใจ ($POS_C(D) = U$) จึงจะเป็นข้อมูลที่เหมาะสมและข้อมูลสมบูรณ์ที่สามารถนำมาสร้างคลาสตัดสินใจที่ดีได้อย่างถูกต้องและเหมาะสม ดังนั้น ถ้าพื้นที่ทางบวกของ $POS_C(D)$ ไม่เท่ากับเซตของแถวข้อมูลทั้งหมดในตารางตัดสินใจเราจะทำการเพิ่มแอททริบิวต์ให้กับแอททริบิวต์เงื่อนไขจนกระทั่งข้อมูลในตารางเป็นข้อมูลสอดคล้องก่อนนำมาค้นหาแอททริบิวต์รีดัก

2. ทฤษฎีรัฟเซตทำให้ทราบว่าแอททริบิวต์ตัวไหนมีความสำคัญ โดยทำการแบ่งแอททริบิวต์ออกเป็น 3 ประเภท ดังนี้ 1. แอททริบิวต์คอร์ 2. แอททริบิวต์รีดัก และ 3. แอททริบิวต์ไม่จำเป็น

นิยามที่ 2.8 แอททริบิวต์รีดัก
กำหนดให้ แอททริบิวต์ a เป็นแอททริบิวต์ไม่จำเป็นใน R ถ้า $POS_R(D) = POS_{R-\{a\}}(D)$ โดยที่ R เป็นเซตย่อยของแอททริบิวต์ C เขียนแทนด้วย $R \subseteq C$ จะกล่าวว่า R' เป็นเซตแอททริบิวต์รีดักของ R

$$\text{ถ้า } POS_R(D) = POS_{R'}(D)$$

โดยที่ แอททริบิวต์ a ทั้งหมดเป็นแอททริบิวต์ไม่จำเป็น ซึ่งเป็นสมาชิกของ R-R' เขียนแทนด้วย $a \in R-R'$

แอททริบิวต์รีดักเป็นเซตแอททริบิวต์เงื่อนไขที่น้อยที่สุดของระบบสารสนเทศที่สามารถแบ่งแยกคลาสตัดสินใจถูกต้องเหมือนกับเซตแอททริบิวต์เงื่อนไขตั้งต้นทั้งหมด ดังนั้นคำตอบแอททริบิวต์รีดักที่เป็นไปได้ทั้งหมดนั้นขึ้นอยู่กับจำนวนแอททริบิวต์เงื่อนไข กล่าวคือ คำตอบแอททริบิวต์รีดักที่เป็นไปได้ทั้งหมด เท่ากับ 2^n-1 โดยที่ n เป็นจำนวนแอททริบิวต์เงื่อนไข

<u>ตัวอย่างที่ 2.5</u> การค้นหาแอททริบิวต์รีดักจากข้อมูลตารางที่ 2.2

จากตัวอย่างที่ 2.3 แสดงให้เห็นว่าข้อมูลในตารางที่ 2.2 สามารถนำมาใช้ค้นหาแอททริบิวต์รีดักได้ เนื่องจากเป็นข้อมูลสอดคล้อง กล่าวคือ $POS_C(D) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$

จากข้อมูลในตารางที่ 2.2 C = {Weight, Door, Size, Cylinder} ดังนั้น คำตอบแอททริบิวต์รีดักที่เป็นไปได้ทั้งหมดมีดังนี้ {Weight}, {Door}, {Size}, {Cylinder}, {Weight, Door}, {Weight, Size}, {Weight, Cylinder}, {Door, Size}, {Door, Cylinder}, {Size, Cylinder}, {Weight, Door, Size}, {Weight, Door, Cylinder}, {Weight, Size, Cylinder}, {Door, Size, Cylinder}, {Weight, Door, Size, Cylinder}

เริ่มทำการค้นหาแอททริบิวต์รีดักโดยการคำนวณพื้นที่ทางบวกของคำตอบแอททริบิวต์รีดักที่เป็นไปได้ทั้งหมด

สมมติให้ R_1 = {Weight} $POS_{R_1}(D) = \{x_3, x_4, x_5, x_7\}$

R_2 = {Door} $POS_{R_2}(D) = \{\}$

R_3 = {Size} $POS_{R_3}(D) = \{x_2, x_7, x_8\}$

R_4 = {Cylinder} $POS_{R_4}(D) = \{x_2, x_4, x_7, x_8\}$

R_5 = {Weight, Door} $POS_{R_5}(D) = \{x_3, x_4, x_5, x_7\}$

R_6 = {Weight, Size} $POS_{R_6}(D) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$

R_7 = {Weight, Cylinder} $POS_{R_7}(D) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$

R_8 = {Door, Size} $POS_{R_8}(D) = \{x_2, x_7, x_8\}$

R_9 = {Door, Cylinder} $POS_{R_9}(D) = \{x_1, x_2, x_4, x_7, x_8\}$

R_{10} = {Size, Cylinder} $POS_{R_{10}}(D) = \{x_2, x_4, x_7, x_8\}$

R_{11} = {Weight, Door, Size} $POS_{R_{11}}(D) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$

R_{12} = {Weight, Door, Cylinder} $POS_{R_{12}}(D) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$

R_{13} = {Weight, Size, Cylinder} $POS_{R_{13}}(D) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$

R_{14} = {Door, Size, Cylinder} $POS_{R_{14}}(D) = \{x_1, x_2, x_4, x_7, x_8\}$

R_{15} = {Weight, Door, Size, Cylinder} $POS_{R_{15}}(D) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$

จากการคำนวณพื้นที่ทางบวกของคำตอบแอททริบิวต์รีดักที่เป็นไปได้แต่ละตัวพบว่า R_1 R_2, R_3, R_4, R_5, R_8, R_9, R_{10}, และ R_{14} ไม่มีคุณสมบัติเป็นแอททริบิวต์รีดัก เนื่องจากเซตแอททริบิวต์ของ R_1 R_2, R_3, R_4, R_5, R_8, R_9, R_{10} และ R_{14} ไม่สามารถแบ่งแยกคลาสตัดสินใจได้ทั้งหมด กล่าวคือ พื้นที่ทางบวกของ R_1 R_2, R_3, R_4, R_5, R_8, R_9, R_{10}, และ R_{14} ไม่เท่ากับเซตออบเจตทั้งหมด 8 ออบเจคในตาราง

เซตแอททริบิวต์ที่มีคุณสมบัติเป็นแอททริบิวต์รีดัก ได้แก่ เซตแอททริบิวต์ R_6 R_7, R_{11}, R_{12}, R_{13}, R_{15} เนื่องจากพื้นที่ทางบวกของเซตแอททริบิวต์เหล่านี้เท่ากับเซตออบเจคทั้งหมด 8 ออบเจคในตาราง ซึ่งจะสังเกตได้ว่าเซตแอททริบิวต์เหล่านี้จะมีแอททริบิวต์ Weight เป็นส่วนหนึ่ง ซึ่งจะสอดคล้องกับตัวอย่างที่ 2.4 คำนวณหาแอททริบิวต์คอร์ได้เท่ากับแอททริบิวต์ Weight ดังนั้น ในบางครั้งการคำนวณหาแอททริบิวต์รีดักจะเริ่มจากการคำนวณหาแอททริบิวต์คอร์ก่อนแล้วจึงนำแอททริบิวต์คอร์มาตรวจสอบว่าเป็นแอททริบิวต์รีดักหรือไม่ ดังนั้นคำตอบแอททริบิวต์รีดักที่จะถูกพิจารณามีดังนี้ {Weight}, {Weight, Door}, {Weight, Size}, {Weight, Cylinder}, {Weight, Door, Size}, {Weight, Door, Cylinder}, {Weight, Size, Cylinder}, {Weight, Door, Size, Cylinder}

แต่จากนิยามแอททริบิวต์รีดักได้กล่าวว่า แอททริบิวต์ R' จะเป็นแอททริบิวต์รีดักถ้า $POS_R(D) = POS_{R'}(D)$ กล่าวคือ R' เป็นเซตแอททริบิวต์ที่น้อยที่สุดที่สามารถแบ่งแยกคลาสตัดสินใจเหมือนกับเซตแอททริบิวต์ตั้งต้นทั้งหมด ดังนั้น เซตแอททริบิวต์ R_{11}, R_{12}, R_{13} และ R_{15} จะไม่เป็นแอททริบิต์รีดักที่เหมาะสมที่สุดหรือเรียกว่าเป็นซุปเปอร์แอททริบิวต์รีดัก เนื่องจากภายในเซตแอททริบิวต์ของ R_{11}, R_{12}, R_{13} และ R_{15} ยังมีเซตแอททริบิวต์ไม่จำเป็น a อยู่ ดังตัวอย่างต่อไปนี้

<u>ตัวอย่างที่ 2.5.1</u> พิจารณาเซตแอททริบิวต์ R_{11} = {Weight, Door, Size} กับ R_6 = {Weight, Size} ซึ่ง R_6 เป็นเซตแอททริบิวต์ย่อยของเซตแอททริบิวต์ R_{11} $(R_6 \subset R_{11})$ และ R_6 มีคุณสมบัติเป็นแอททริบิวต์รีดักเหมือนกับเซตแอททริบิวต์ R_{11} แสดงว่าเซตแอททริบิวต์ R_{11} ยังคงมีเซตแอททริบิวต์ไม่จำเป็น a ประกอบอยู่ด้วย นั่นคือ แอททริบิวต์ไม่จำเป็น $a = R_{11} - R_6$ = {Weight, Door, Size}-{Weight, Size} = {Door} แสดงว่า a = {Door} เป็นแอททริบิวต์ไม่จำเป็นทั้งหมดใน R_{11} ดังนั้น R_{11} = {Weight, Door, Size} ไม่เป็นแอททริบิวต์รีดัก

<u>ตัวอย่างที่ 2.5.2</u> พิจารณาเซตแอททริบิวต์ R_{12} = {Weight, Door, Cylinder} กับ R_7 = {Weight, Cylinder} ซึ่ง R_7 เป็นเซตแอททริบิวต์ย่อยของเซตแอททริบิวต์ R_{12} $(R_7 \subset R_{12})$ และ R_7 มีคุณสมบัติเป็นแอททริบิวต์รีดักเหมือนกับเซตแอททริบิวต์ R_{12} แสดงว่าเซตแอททริบิวต์ R_{12} ยังคงมีเซตแอททริบิวต์ไม่จำเป็น a ประกอบอยู่ด้วย นั่นคือ แอททริบิวต์ไม่จำเป็น $a = R_{12} - R_7$ = {Weight, Door, Cylinder}-{Weight, Cylinder} = {Door} แสดงว่า a = {Door} เป็นแอททริบิวต์ไม่จำเป็นทั้งหมดใน R_{12} ดังนั้น R_{12} = {Weight, Door, Cylinder} ไม่เป็นแอททริบิวต์รีดัก

<u>ตัวอย่างที่ 2.5.3</u> พิจารณาเซตแอททริบิวต์ R_{13} = {Weight, Size, Cylinder} กับ R_6 = {Weight, Size} ซึ่ง R_6 เป็นเซตแอททริบิวต์ย่อยของเซตแอททริบิวต์ R_{13} $(R_6 \subset R_{13})$ และ R_6 มีคุณสมบัติเป็นแอททริบิวต์รีดักเหมือนกับเซตแอททริบิวต์ R_{13} แสดงว่าเซตแอททริบิวต์ R_{13} ยังคงมีเซตแอททริบิวต์ไม่จำเป็น a ประกอบอยู่ด้วย นั่นคือ แอททริบิวต์ไม่จำเป็น a = $R_{13} - R_6$ = {Weight, Size, Cylinder} – {Weight, Size} = {Cylinder} แสดงว่า a = {Cylinder} เป็นแอททริบิวต์ไม่จำเป็นทั้งหมดใน R_6 ดังนั้น R_6 = {Weight, Size, Cylinder} ไม่เป็นแอททริบิวต์รีดัก

<u>ตัวอย่างที่ 2.5.4</u> พิจารณาเซตแอททริบิวต์ R_{13} = {Weight, Size, Cylinder} กับ R_7 = {Weight, Cylinder} ซึ่ง R_7 เป็นเซตแอททริบิวต์ย่อยของเซตแอททริบิวต์ R_{13} $(R_7 \subset R_{13})$ และ R_7 มีคุณสมบัติเป็นแอททริบิวต์รีดักเหมือนกับเซตแอททริบิวต์ R_{13} แสดงว่าเซตแอททริบิวต์ R_{13} ยังคงมีเซตแอททริบิวต์ไม่จำเป็น a ประกอบอยู่ด้วย นั่นคือ แอททริบิวต์ไม่จำเป็น a = $R_{13} - R_7$ = {Weight, Door, Cylinder} - {Weight, Cylinder} = {Door} แสดงว่า a = {Door} เป็นแอททริบิวต์ไม่จำเป็นทั้งหมดใน R_{13} ดังนั้น R_{13} = {Weight, Size, Cylinder} ไม่เป็นแอททริบิวต์รีดัก

<u>ตัวอย่างที่ 2.5.5</u> พิจารณาเซตแอททริบิวต์ R_{15} = {Weight, Door, Size, Cylinder} กับ R_6 = {Weight, Size} ซึ่ง R_6 เป็นเซตแอททริบิวต์ย่อยของเซตแอททริบิวต์ R_{15} $(R_6 \subset R_{15})$ และ R_6 มีคุณสมบัติเป็นแอททริบิวต์รีดักเหมือนกับเซตแอททริบิวต์ R_{15} แสดงว่าเซตแอททริบิวต์ R_{15} ยังคงมีเซตแอททริบิวต์ไม่จำเป็น a ประกอบอยู่ด้วย นั่นคือ แอททริบิวต์ไม่จำเป็น a = $R_{15} - R_6$ = {Weight, Door, Size, Cylinder} – {Weight, Size} = {Door, Cylinder} แสดงว่า a = {Door, Cylinder} เป็นแอททริบิวต์ไม่จำเป็นทั้งหมดใน R_{15} ดังนั้น R_{15} = {Weight, Door, Size, Cylinder} ไม่เป็นแอททริบิวต์รีดัก

<u>ตัวอย่างที่ 2.5.6</u> พิจารณาเซตแอททริบิวต์ R_{15} = {Weight, Door, Size, Cylinder} กับ R_7 = {Weight, Cylinder} ซึ่ง R_7 เป็นเซตแอททริบิวต์ย่อยของเซตแอททริบิวต์ R_{15} $(R_7 \subset R_{15})$ และ R_7 มีคุณสมบัติเป็นแอททริบิวต์รีดักเหมือนกับเซตแอททริบิวต์ R_{15} แสดงว่าเซตแอททริบิวต์ R_{15} ยังคงมีเซตแอททริบิวต์ไม่จำเป็น a ประกอบอยู่ด้วย นั่นคือ แอททริบิวต์ไม่จำเป็น a = $R_{15} - R_7$ = {Weight, Door, Size, Cylinder} - {Weight, Cylinder} = {Door, Size} แสดงว่า a = {Door, Size} เป็นแอททริบิวต์ไม่จำเป็นทั้งหมดใน R_{15} ดังนั้น R_{15} = {Weight, Door, Size, Cylinder} ไม่เป็นแอททริบิวต์รีดัก

สรุปได้ว่าเมื่อทำการพิจารณาเซตแอททริบิวต์ R_{11}, R_{12}, R_{13} และ R_{15} พบว่ามีเซตแอททริบิวต์ไม่จำเป็นประกอบด้วยอยู่จริง ทำให้เซตแอททริบิวต์ R_{11}, R_{12}, R_{13} และ R_{15} ไม่เป็นแอททริบิวต์รีดัก และจากตัวอย่างที่ 2.5.3 และตัวอย่างที่ 2.5.4 ได้ทำการพิจารณาเซตแอททริบิวต์ R_{13} ซึ่งจากตัวอย่างทั้งสองแสดงให้เห็นว่าบางครั้งแอททริบิวต์หนึ่งอาจจะเป็นส่วนหนึ่งของแอททริบิวต์รีดักแต่บางสถานการณ์ไม่จำเป็น เช่นเดียวกันจากตัวอย่างที่ 2.5.5 และตัวอย่างที่ 2.5.6 ได้ทำการพิจารณาเซตแอททริบิวต์ R_{15} ซึ่งจากตัวอย่างดังกล่าวแสดงให้เห็นว่าบางครั้งแอททริบิวต์หนึ่งอาจจะเป็นส่วนหนึ่งของแอททริบิวต์รีดักแต่บางสถานการณ์ไม่จำเป็น ดังนั้น เซตแอททริบิวต์รีดักนั้นจะขึ้นอยู่กับการนำแอททริบิวต์ใดบ้างมาประกอบรวมกันเป็นแอททริบิวต์รีดัก

ดังนั้น จากตัวอย่างที่ 2.5 จะสรุปได้ว่าส่วนคำตอบแอททริบิวต์รีดัก R_6 และ R_7 มีคุณสมบัติเป็นแอททริบิวต์รีดักและเป็นเซตแอททริบิวต์ที่น้อยที่สุดที่สามารถแบ่งแยกคลาสตัดสินใจได้ทั้งหมดเหมือนกับเซตแอททริบิวต์เงื่อนไขตั้งต้นทั้งหมด เนื่องจากไม่มีเซตแอททริบิวต์ย่อยทั้งหมดของ R_6 และ R_7 ที่มีคุณสมบัติเป็นแอททริบิวต์รีดัก ยกตัวอย่างเช่น เซตแอททริบิวต์ R_1 = {Weight} กับ R_3 = {Size} ซึ่งเป็นเซตย่อยทั้งหมดของ R_6 = {Weight, Size} และ R_1 = {Weight} กับ R_3 = {Size} ไม่มีคุณสมบัติเป็นแอททริบิวต์รีดัก เช่นเดียวกันกับเซตแอททริบิวต์ R_1 = {Weight} กับ R_4 = {Cylinder} ซึ่งเป็นเซตย่อยทั้งหมดของ R_7 = {Weight, Cylinder} และ R_1 = {Weight} กับ R_4 = {Cylinder} ไม่มีคุณสมบัติเป็นแอททริบิวต์รีดัก ดังนั้นสรุปได้ว่า แอททริบิวต์รีดักที่เหมาะสมที่สุดหรือแอททริบิวต์รีดักที่น้อยที่สุด คือ เซตแอททริบิวต์ R_6 = {Weight, Size} และ R_7 = {Weight, Cylinder}

จากเซตคำตอบแอททริบิวต์รีดักที่เป็นไปได้ทั้งหมดจะสามารถสรุปได้ว่าแต่ละคำตอบแอททริบิวต์รีดักจะถูกแบ่งออกเป็น 3 ประเภท ดังนี้ 1. เซตแอททริบิวต์ไม่เป็นแอททริบิวต์รีดัก 2. เซตแอททริบิวต์เป็นซุปเปอร์แอททริบิวต์รีดัก และ 3. เซตแอททริบิวต์เป็นแอททริบิวต์รีดักที่เหมาะสมที่สุดหรือกล่าวได้ว่าเป็นแอททริบิวต์ที่น้อยที่สุดตามนิยามแอททริบิวต์รีดัก

นอกจากนี้จากตัวอย่างจะสังเกตเห็นว่า ถ้าค้นพบเซตแอททริบิวต์ใดเป็นแอททริบิวต์รีดักที่น้อยที่สุดแล้วเซตแอททบิวต์อื่นที่นำมาประกอบเข้าร่วมกับเซตแอททริบิวต์รีดักนี้ก็จะมีคุณสมบัติเป็นแอททริบิวต์รีดักได้ด้วยเช่นกัน ยกตัวอย่างเช่น ถ้าเซตแอททริบิวต์ R_6 = {Weight, Size} เป็นแอททริบิวต์รีดัก แสดงว่า ถ้านำเซตแอททริบิวต์ R_6 = {Weight, Size} บวกร่วมกับเซตแอททริบิวต์อื่น เช่น R_6 + {Door} = {Weight, Size, Door}, R_6 + {Cylinder} = {Weight, Size, Cylinder} และ R_6 + {Door, Cylinder} = {Weight, Size, Door, Cylinder} จะพบว่าเซตแอททริบิวต์ {Weight, Size, Door}, {Weight, Size, Cylinder} และ {Weight, Size, Door, Cylinder} มีคุณสมบัติเป็น

แอททริบิวต์รีดักด้วยเช่นกัน หรือกล่าวอีกนัยนึงว่า ถ้าค้นพบเซตแอททริบิวต์ที่มีคุณสมบัติเป็นแอททริบิวต์รีดัก จะบอกได้ว่าจะต้องมีเซตแอททริบิวต์ย่อยของเซตแอททริบิวต์เหล่านั้นเป็นแอททริบิวต์รีดัก ยกตัวอย่างเช่น R_{11}, R_{12}, R_{13} และ R_{15} เป็นเซตแอททริบิวต์ที่มีคุณสมบัติเป็นแอททริบิวต์รีดัก ดังนั้น แสดงว่ามีเซตแอททริบิวต์ย่อยของ R_{11}, R_{12}, R_{13} และ R_{15} เป็นแอททริบิวต์รีดัก

จากการค้นหาแอททริบิวต์รีดักในบางครั้งอาจจะพบคำตอบแอททริบิวต์รีดักมากกว่า 1 คำตอบ ซึ่งโดยธรรมชาติแล้วไม่มีใครทราบว่าเซตแอททริบิวต์รีดักตัวไหนดีเหมาะสมที่สุด ซึ่งจะขึ้นอยู่กับว่าเราสนใจเซตแอททริบิวต์รีดักเซตใดเพื่อนำไปใช้ในการทำนายคลาสตัดสินใจ

สรุปขั้นตอนการหาแอททริบิวต์รีดักมี 2 ขั้นตอน ดังนี้ ขั้นแรกทำการตรวจสอบข้อมูลว่าเป็นข้อมูลสอดคล้อง ($POS_C(D) = U$) หรือไม่ และขั้นตอนที่ 2 ทำการค้นหาแอททริบิวต์รีดักจากคำตอบแอททริบิวต์รีดักที่เป็นไปได้ทั้งหมดซึ่งขึ้นอยู่กับจำนวนแอททริบิวต์เงื่อนไข

ดังนั้น จากนิยามแอททริบิวต์รีดักในทฤษฎีรัฟเซต จะกล่าวได้ว่า R' เป็นแอททริบิวต์รีดักในตารางตัดสินใจ ถ้า $POS_{R'}(D) = POS_C(D)$ และ $\forall B \subset R'; POS_B(D) \neq POS_C(D)$ โดยที่ $R' \subseteq C$ ซึ่งจากคุณสมบัติของทฤษฎีรัฟเซตได้กล่าวว่าข้อมูลตั้งต้นก่อนค้นหาแอททริบิวต์รีดักนั้นจะเป็นข้อมูลสอดคล้อง กล่าวคือ $POS_C(D) = U$

ดังนั้น จะสรุปนิยามแอททริบิวต์รีดักในทฤษฎีรัฟเซตได้ว่า R' เป็นแอททริบิวต์รีดักในตารางตัดสินใจ ถ้า $POS_{R'} = U$ และ $\forall B \subset R'; POS_B(D) \neq U$ โดยที่ $R' \subseteq C$

นิยามที่ 2.6 ตารางเมตริกมองเห็นได้ (Discernibility Matrix) และฟังก์ชันมองเห็นได้ (Discernibility Function)

ตารางเมตริกมองเห็นได้ M(T) ของตารางตัดสินใจ $T = (U, C \cup D, V, f)$ คือ เมตริกของตารางตัดสินใจ $|U| \times |U|$, โดยที่ m_{ij} เป็นเซตแอททริบิวต์ทั้งหมดที่สามารถแบ่งแยกความแตกต่างระหว่างออบเจค i และออบเจค j ของคลาสตัดสินใจที่แตกต่างกันใน U/IND(D) เขียนแทนด้วย

$$\left.\begin{array}{l} m_{ij} \\ \end{array}\right\} \begin{array}{l} = \{c : f(x_i, C) \neq f(x_j, C) ; f(x_i, d) \neq f(x_j, d)\} \\ = \{\lambda ; f(x_i, d) = f(x_j, d)\} \text{ ซึ่ง } c \in C \text{ และ } d \in D \text{ สำหรับ } i, j = 1, 2, ..., n \end{array}$$

F(M) เป็นฟังก์ชันมองเห็นได้ที่แสดงถึงแอททริบิวต์รีดักทั้งหมดของตารางตัดสินใจ เขียนแทนด้วย

$$F(M) = \wedge_{1 \leq j \leq i \leq n} \{\vee m_{ij}\}$$

ตัวอย่างที่ 2.6 การค้นหาแอททริบิวต์รีดักจากการสร้างตารางเมตริกมองเห็นได้

จากข้อมูลในตารางที่ 2.6 นำมาสร้างตารางเมตริกมองเห็นได้ตามนิยามที่ 2.6 จะได้ผลดังตารางที่ 2.7

ตารางที่ 2.6 ตารางตัดสินใจ

U	a	b	c	d	D
x_1	1	0	2	1	1
x_2	1	0	2	0	1
x_3	1	2	0	0	2
x_4	1	2	2	1	0
x_5	2	1	0	0	2
x_6	2	1	1	0	2
x_7	2	1	2	1	1

ตารางที่ 2.7 ตารางเมตริกมองเห็นได้

U	x_1	x_2	x_3	x_4	x_5	x_6	F(M)
x_2	λ						λ
x_3	b, c, d	b, c					(b $\vee$ c)
x_4	b	b, d	c, d				b$\wedge$ (c $\vee$ d)
x_5	a, b, c, d	a, b, c	λ	a, b, c, d			(a $\vee$ b $\vee$ c)
x_6	a, b, c, d	a, b, c	λ	a, b, c, d	λ		(a $\vee$ b $\vee$ c)
x_7	λ	λ	a, b, c, d	a, b	c, d	c, d	(a $\vee$ b) $\wedge$ (c $\vee$ d)

แอททริบิวต์รีดัก = $\lambda \wedge$(b $\vee$ c) $\wedge$ b$\wedge$ (c $\vee$ d) $\wedge$(a $\vee$ b $\vee$ c) $\wedge$(a $\vee$ b $\vee$ c) $\wedge$(a $\vee$ b) $\wedge$ (c $\vee$ d)

= b $\wedge$ (c $\vee$ d)

= {b, c}, {b, d}

2.3 วิธีการคำนวณหาแอททริบิวต์รีดัก

การคำนวณหาแอททริบิวต์รีดักในทฤษฎีรัฟเซตนั้นมี 2 วิธี ได้แก่ วิธีฟังก์ชันมองเห็นได้ และวิธีพื้นที่ทางบวกรายละเอียดแต่ละวิธีพร้อมทั้งงานวิจัยที่เกี่ยวข้องจะได้กล่าวถึงต่อไป

2.3.1 วิธีฟังก์ชันมองเห็นได้

ในการค้นหาแอททริบิวต์รีดักที่น้อยที่สุดและแอททริบิวต์รีดักทั้งหมดโดยใช้ฟังก์ชันมองเห็นได้นั้นเป็นปัญหาเอ็นพีฮาร์ด (NP-hard) กล่าวคือ ในการค้นหาแอททริบิวต์รีดักจะต้องทำการสร้างตารางเมตริกจากนั้นเปรียบเทียบระหว่างคู่แถวข้อมูลที่เป็นไปได้ทั้งหมด ซึ่งจะทำให้ใช้เนื้อที่ในหน่วยความจำขนาดใหญ่สำหรับตารางเมตริก และใช้เวลาในการเปรียบเทียบคู่แถวเป็นจำนวนมาก ดังนั้น แม้ว่าวิธีนี้จะคำนวณได้ง่าย แต่วิธีนี้จะไม่เหมาะสมกับข้อมูลที่มีขนาดใหญ่เนื่องจากใช้เวลาในการคำนวณและเนื้อที่ในหน่วยความจำเป็นจำนวนมาก

จากปัญหาดังกล่าวจึงได้มีงานวิจัยมาช่วยปรับปรุงโดยเสนอการค้นหาแบบฮิวริสติก ดังนี้

Keyun และคณะ [7] เสนออัลกอริทึมค้นหาแอททริบิวต์รีดักแบบฮิวริสติกเพื่อทำให้การค้นหาแอททริบิวต์รีดักทำงานได้เร็วขึ้น โดยใช้ความถี่ของแอททริบิวต์ที่ปรากฏในตารางเมตริกเป็นฟังก์ชันฮิวริสติกเพื่อช่วยในการเลือกแอททริบิวต์ ซึ่งผลลัพธ์ที่ได้รับจากการใช้วิธีนี้แสดงให้เห็นว่าทำงานได้เร็วขึ้น แต่แอททริบิวต์รีดักที่ได้รับโดยส่วนใหญ่แล้วเป็นซุปเปอร์เซตแอททริบิวต์รีดักไม่ใช่แอททริบิวต์รีดักที่เหมาะสมที่สุด

Keyun และคณะ [8] เสนออัลกอริทึมค้นหาแอททริบิวต์รีดักสำหรับข้อมูลขนาดใหญ่ โดยใช้ความถี่ของแอททริบิวต์ที่ปรากฏในตารางเมตริกเป็นฟังก์ชันฮิวริสติกเพื่อช่วยในการเลือกแอททริบิวต์ร่วมกับการใช้วิธีการสุ่ม (Sampling) เพื่อใช้ในการสุ่มตัวอย่างข้อมูลจากข้อมูลขนาดใหญ่ทั้งหมด ซึ่งผลลัพธ์ที่ได้รับจากการใช้วิธีนี้ แสดงให้เห็นว่าแอททริบิวต์รีดักที่ได้รับนั้นเป็นเพียงการประมาณแอททริบิวต์รีดักไม่ใช่แอททริบิวต์รีดักที่เหมาะสมที่สุด นอกจากนี้อัลกอริทึมมีการทำงานที่ซับซ้อน เนื่องจากมีการสุ่มตัวอย่างหลายครั้งและในแต่ละครั้งของการสุ่มตัวอย่างจะมีการสร้างตารางเมตริกสำหรับข้อมูลนั้น จากนั้นทำการคำนวณความถี่ของแอททริบิวต์ที่ปรากฏในตารางขึ้นมาใหม่เพื่อทำการค้นหาแอททริบิวต์รีดัก แล้วทำการตรวจสอบว่าแอททริบิวต์รีดักที่ได้รับในแต่ละการสุ่มตัวอย่างเท่ากันหรือไม่ ซึ่งจะทำเช่นนี้จนกว่าจะค้นหาแอททริบิวต์รีดักที่แน่นอน

Zhang และคณะ [5] เสนออัลกอริทึมค้นหาแอททริบิวต์รีดักโดยใช้ความถี่ของแอททริบิวต์ที่ปรากฏในตารางเมตริกร่วมกับความยาวของแอททริบิวต์เป็นฟังก์ชันฮิวริสติก ซึ่งในการพิจารณาแอททริบิวต์จากคุณสมบัติทั้งสอง คือ จะเลือกเซตย่อยแอททริบิวต์ที่มีความถี่สูงสุด และถ้ามีหลาย ๆ เซตย่อยแอททริบิวต์มีความถี่เท่ากันจะเลือกเซตย่อยแอททริบิวต์ที่มีความยาวรายการที่น้อยที่สุด เพื่อช่วยให้แอททริบิวต์ที่ได้รับจากคุณสมบัตินี้เป็นแอททริบิวต์ที่มีความตรงประเด็นและไม่ซ้ำซ้อน ดังนั้น ผลลัพธ์ที่ได้รับจากอัลกอริทึมนี้ แสดงให้เป็นว่าแอททริบิวต์รีดักที่ได้รับเป็นแอททริบิวต์ที่เหมาะสมที่สุดและยังช่วยปรับปรุงให้ทำงานได้เร็วขึ้น อย่างไรก็ตาม อัลกอริทึมนี้มีการทำงานที่ซับซ้อนเนื่องจากในแต่ละครั้งการเลือกแอททริบิวต์ จะทำการตรวจสอบว่าแอททริบิวต์นั้นเหมาะสมที่จะเป็นแอททริบิวต์รีดักหรือไม่ ถ้าไม่เหมาะสมก็จะทำการคำนวณความถี่และความยาวของรายการใหม่หมด กล่าวคือ เมื่อแอททริบิวต์ใดถูกเลือกเป็นแอททริบิวต์รีดักแล้วแอททริบิวต์ที่ถูกเลือกนั้นจะถูกกำจัดออกจากตารางเมตริก แล้วทำการคำนวณความถี่และความยาวของรายการใหม่ จากนั้นทำการเลือกแอททริบิวต์จากคุณสมบัติทั้งสองเพื่อนำแอททริบิวต์ที่ถูกเลือกบวกเข้ากับแอททริบิวต์รีดักเดิม แล้วตรวจสอบว่าเซตแอททริบิวต์รีดักนี้เหมาะสมหรือไม่ จะทำเช่นนี้จนกว่าเซตแอททริบิวต์ที่ถูกเลือกนี้เป็นแอททริบิวต์รีดักเหมาะสมที่สุด

2.3. 2 วิธีพื้นที่ทางบวก

วิธีนี้จะทำการสร้างคลาสสมมูลกันซึ่งขึ้นอยู่กับค่าแอททริบิวต์เงื่อนไขกับแอททริบิวต์ตัดสินใจเพื่อคำนวณพื้นที่ทางบวกจากค่าประมาณขอบเขตล่างซึ่งทำให้คำนวณยุ่งยากและจะต้องพิจารณาทุก ๆ เซตย่อยของแอททริบิวต์เงื่อนไขที่เป็นไปได้ทั้งหมดทำให้เสียเวลามาก แม้ว่าวิธีนี้ให้แอททริบิวต์รีดักที่เหมาะสมที่สุด แต่จัดได้ว่าเป็นวิธีที่ไม่มีประสิทธิภาพสำหรับข้อมูลขนาดใหญ่

จากปัญหาดังกล่าวจึงได้มีงานวิจัยมาช่วยปรับปรุงโดยเสนอวิธีการคำนวณหาแอททริบิวต์รีดักใหม่ ดังนี้

Xiaohua Tony Hu และคณะ [11] เสนอโมเดลรัฟเซตใหม่โดยใช้รีเลชันนอลแอลจีบราโอเปอร์เรชัน (Relaitonal Algebra Operation) มากำหนดนิยามแอททริบิวต์คอร์และแอททริบิวต์รีดักในทฤษฎีรัฟเซตขึ้นใหม่และสร้างอัลกอริทึมตามนิยามแอททริบิวต์คอร์และแอททริบิวต์รีดักที่สร้างขึ้นเพื่อให้คำนวณหาแอททริบิวต์คอร์และแอททริบิวต์รีดักได้ง่าย นอกจากนี้ยังสามารถทำงานได้ดีสำหรับข้อมูลที่มีขนาดใหญ่เนื่องจากรีเลชันนอลแอลจีบราโอเปอร์เรชันอยู่บนระบบฐานข้อมูลอย่างแพร่หลาย และในปัจจุบันข้อมูลส่วนใหญ่จะถูกเก็บอยู่ในรูปฐานข้อมูลมากกว่าในรูปไฟล์ข้อมูล จึงทำให้นำไปประยุกต์ใช้งานได้ง่าย ดังนั้น โมเดลรัฟเซตใหม่นี้จะสามารถคำนวณหาแอททริบิวต์รีดักได้ง่ายและเหมาะสมกับข้อมูลขนาดใหญ่ อย่างไรก็ตามอัลกอริทึมนี้ให้เซตแอททริบิวต์ที่เป็นแอททริบิวต์รีดักเพียงคำตอบเดียว เนื่องจากถ้าค้นหาแอททริบิวต์รีดักที่เป็นไปได้ทั้งหมดจะต้องตรวจสอบเซตย่อยแอททริบิวต์เงื่อนไขทั้งหมดที่สามารถเป็นเซตแอททิบิวต์รีดักได้ ซึ่งจะทำให้เสียเวลา และในความเป็นจริงก็ไม่ทราบว่าแอททริบิวต์รีดักไหนดีที่สุด กล่าวคือ เซตแอททริบิวต์รีดักหนึ่งสามารถนำไปใช้ได้กับบางสถานะการณ์เท่านั้น

Jianchao และคณะ [4] เสนอนิยามการขึ้นต่อกันของแอททริบิวต์ใหม่ซึ่งเรียกว่าการขึ้นต่อกันระหว่างแอททริบิวต์ที่สัมพันธ์ เพื่อนำมาใช้ในการช่วยตรวจสอบว่าเซตแอททริบิวต์เป็นเซตแอททริบิวต์รีดักหรือไม่แทนที่การคำนวณจากวิธีพื้นที่ทางบวก โดยการคำนวณได้จากการนับความแตกต่างของแถวทำให้คำนวณได้อย่างมีประสิทธิภาพ นอกจากนี้ยังสร้างอัลกอริทึมสำหรับการค้นหาแอททริบิวต์รีดักที่น้อยที่สุด 2 อัลกอริทึม ดังนี้ 1. บรูทฟอดอัลกอริทึม (Brute-Force Algorithm) ในการคำนวณจะทำการประเมินแอททริบิวต์ทั้งหมดในเซตแอททริบิวต์เงื่อนไข และ 2. ฮิวริสติกอัลกอริทึม โดยใช้เอนโทรปีแอททริบิวต์ (Attribute Entropy) เป็นฟังก์ชันฮิวริสติกเพื่อนำมาช่วยเรียงลำดับความสำคัญของแอททริบิวต์ทำให้ได้รับแอททริบิวต์รีดักที่น้อยที่สุดได้รวดเร็วขึ้น แต่จากอัลกอริทึมทั้ง 2 อัลกอริทึมสามารถหาแอททริบิวต์รีดักได้เพียงเซตแอททริบิวต์เดียว นั่นคือ ไม่สามารถหาเซตแอททริบิวต์รีดักที่น้อยที่สุดที่เป็นไปได้ทั้งหมด

บทที่ 3
แอททริบิวต์รีดักบนฟังก์ชันการขึ้นต่อกัน

ในบทที่ 3 นี้ได้นำเสนอนิยามแอททริบิวต์รีดักบนฟังก์ชันการขึ้นต่อกัน โดยแบ่งเนื้อหาออกเป็น 4 ส่วน ดังนี้ ส่วนแรกจะกล่าวถึงทฤษฎีฟังก์ชันการขึ้นต่อกัน ส่วนที่ 2 จะกล่าวถึงคุณสมบัติของฟังก์ชันการขึ้นต่อกัน ส่วนที่ 3 จะกล่าวถึงการวิเคราะห์นิยามแอททริบิวต์รีดัก ส่วนที่ 4 จะกล่าวถึงนิยามแอททริบิวต์รีดักบนทฤษฎีฟังก์ชันการขึ้นต่อกัน และตัวอย่างแอททริบิวต์รีดักตามนิยามแอททริบิวต์รีดักบนฟังก์ชันการขึ้นต่อกัน

3.1 ภาพรวมของทฤษฎีฟังก์ชันการขึ้นต่อกัน

นิยามพื้นฐานของทฤษฎีฟังก์ชันการขึ้นต่อกัน [9, 12, 13, 14, 15] มีรายละเอียด ดังนี้

นิยามที่ 3.1 ฟังก์ชันการขึ้นต่อกัน (Functional Dependency : FD)
ฟังก์ชันการขึ้นต่อกันเป็นความสัมพันธ์ระหว่างเซตแอททริบิวต์ในตารางฐานข้อมูลเชิงสัมพันธ์ R, กำหนดให้ A และ X เป็นเซตแอททริบิวต์ในตารางฐานข้อมูลเชิงสัมพันธ์ และ r เป็นค่าข้อมูลของเซตแอททริบิวต์ในตารางฐานข้อมูลเชิงสัมพันธ์ จะกล่าวว่า แอททริบิวต์ A บ่งชี้แอททริบิวต์ X เขียนแทนด้วย $A \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกัน ก็ต่อเมื่อ สำหรับทุก ๆ คู่แถวที่ t_i และ t_j ใน r ถ้าแถวที่ t_i และ t_j มีค่าของเซตแอททริบิวต์ A ทั้งหมดเท่ากันแล้ว แถวที่ t_i และ t_j มีค่าของเซตแอททริบิวต์ X ทั้งหมดเท่ากันด้วย นั่นคือ แต่ละค่าของแอททริบิวต์ A สามารถบ่งชี้ค่าแอททริบิวต์ X ได้เพียงค่าเดียวเสมอ เขียนแทนด้วย

$$\forall (t_i, t_j) : \text{ถ้า } t_i(A) = t_j(A) \text{ แล้ว } t_i(X) = t_j(X)$$

กำหนดให้ $A \longrightarrow X$ จะกล่าวได้ว่า เซตแอททริบิวต์ A เป็นข้างซ้าย (Left Hand Side : LHS) ของฟังก์ชันขึ้นต่อกัน ซึ่งเรียกว่า ดีเทอมิแนนท์ (Determinant) และเซตแอททริบิวต์ X เป็นข้างขวา (Right Hand Side : RHS) ของฟังก์ชันขึ้นต่อกัน ซึ่งเรียกว่า ออบเจคของดีเทอมิแนนท์ (Object Of The Determinant)

<u>ตัวอย่างที่ 3.1</u> แอททริบิวต์ A บ่งชี้แอททริบิวต์ X เป็นฟังก์ชันการขึ้นต่อกัน ($A \longrightarrow X$)

จากตารางที่ 3.1 แสดงให้เห็นว่าแต่ละค่าของแอททริบิวต์ A สามารถบ่งชี้ค่าแอททริบิวต์ X ได้เพียงค่าเดียวเสมอ เนื่องจาก

แถวที่ t_1, t_7 มีค่าแอททริบิวต์ A เท่ากับ 0 เท่ากันแล้วแถวที่ t_1, t_7 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 1 เท่ากัน

แถวที่ t_2, t_4, t_6 มีค่าแอททริบิวต์ A เท่ากับ 1 เท่ากันแล้วแถวที่ t_2, t_4, t_6 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 2 เท่ากัน

แถวที่ t_3, t_5 มีค่าแอททริบิวต์ A เท่ากับ 2 เท่ากันแล้วแถวที่ t_3, t_5 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 3 เท่ากัน

ดังนั้น แอททริบิวต์ A บ่งชี้แอททริบิวต์ X เป็นฟังก์ชันการขึ้นต่อกัน

ตารางที่ 3.1 ตารางฐานข้อมูลเชิงสัมพันธ์

Tuple	A	B	X
t_1	0	2	1
t_2	1	2	2
t_3	2	2	3
t_4	1	1	2
t_5	2	2	3
t_6	1	1	2
t_7	0	1	1

<u>ตัวอย่างที่ 3.2</u> แอททริบิวต์ B บ่งชี้แอททริบิวต์ X ไม่เป็นฟังก์ชันการขึ้นต่อกัน ($B \not\longrightarrow X$)

จากตารางที่ 3.1 แสดงให้เห็นว่าแต่ละค่าของแอททริบิวต์ B ไม่สามารถบ่งชี้ค่าแอททริบิวต์ X ได้เพียงค่าเดียวเสมอ หรือกล่าวได้ว่า ค่าของแอททริบิวต์ B หนึ่งค่าบ่งชี้ค่าแอททริบิวต์ X ได้หลายค่า ซึ่งเรียกเหตุการณ์นี้ว่าวันทูเมนี่ (one to many) เนื่องจาก

แถวที่ t_1, t_2, t_3, t_5 มีค่าแอททริบิวต์ B เท่ากับ 2 เท่ากันแต่แถวที่ t_1, t_2, t_3, t_5 มีค่าแอททริบิวต์ X ไม่เท่ากัน เนื่องจาก แถวที่ t_1 มีค่าแอททริบิวต์ X เท่ากับ 1 แต่แถวที่ t_2 มีค่าแอททริบิวต์ X เท่ากับ 2 และแถวที่ t_3, t_5 มีค่าแอททริบิวต์ X เท่ากับ 3

แถวที่ t_4, t_6, t_7 มีค่าแอททริบิวต์ B เท่ากับ 1 เท่ากันแต่แถวที่ t_4, t_6, t_7 มีค่าแอททริบิวต์ X ไม่เท่ากัน เนื่องจาก t_4, t_6 มีค่าแอททริบิวต์ X เท่ากับ 2 แต่แถวที่ t_7 มีค่าแอททริบิวต์ X เท่ากับ 1

ดังนั้น แอททริบิวต์ B บ่งชี้แอททริบิวต์ X ไม่เป็นฟังก์ชันการขึ้นต่อกัน

ฟังก์ชันการขึ้นต่อกันสามารถแบ่งออกเป็น 2 ประเภท ดังนี้

1. ฟังก์ชันการขึ้นต่อกันแบบบางส่วน (Partial Functional Dependency : PFD)
2. ฟังก์ชันการขึ้นต่อกันแบบเต็ม (Full Functional Dependency : FFD)

นิยามที่ 3.2 ฟังก์ชันการขึ้นต่อกันแบบบางส่วน

กำหนดให้ $A \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกัน จะกล่าวว่า แอททริบิวต์ A บ่งชี้แอททริบิวต์ X เป็นฟังก์ชันการขึ้นต่อกันแบบบางส่วน เขียนแทนด้วย $A \xrightarrow[P]{} X$ ก็ต่อเมื่อ

$$\exists B \subset A \; ; \; B \longrightarrow X$$

กำหนดให้ $A \xrightarrow[P]{} X$ จะกล่าวได้ว่า เซตแอททริบิวต์ A เป็นข้างซ้ายของฟังก์ชันขึ้นต่อกันแบบบางส่วน ซึ่งเรียกว่า ดีเทอมิแนนท์บางส่วน (Partial Determinant) และเซตแอททริบิวต์ X เป็นข้างขวาของฟังก์ชันขึ้นต่อกันบางส่วน ซึ่งเรียกว่า ออบเจคของดีเทอมิแนนท์บางส่วน

<u>ตัวอย่างที่ 3.3</u> เซตแอททริบิวต์ AB บ่งชี้แอททริบิวต์ X เป็นฟังก์ชันการขึ้นต่อกันแบบบางส่วน

จากตารางที่ 3.2 แสดงให้เห็นว่าแต่ละค่าของเซตแอททริบิวต์ AB บ่งชี้ค่าแอททริบิวต์ X ได้เพียงค่าเดียวเสมอ เนื่องจาก

แถวที่ t_1 กับ t_7 มีค่าเซตแอททริบิวต์ A และ B เท่ากับ 0 และ 1 ตามลำดับ ซึ่งเท่ากันแล้วแถวที่ t_1 กับ t_7 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 1 เท่ากันด้วย

แถวที่ t_2 มีค่าเซตแอททริบิวต์ A และ B เท่ากับ 1 และ 2 ตามลำดับ แล้วแถวที่ t_2 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 2 เพียงค่าเดียว

แถวที่ t_3 กับ t_5 มีค่าเซตแอททริบิวต์ A และ B เท่ากับ 2 และ 2 ตามลำดับ ซึ่งเท่ากันแล้วแถวที่ t_3 กับ t_5 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 3 เท่ากันด้วย

แถวที่ t_4 กับ t_6 มีค่าเซตแอททริบิวต์ A และ B เท่ากับ 1 และ 1 ตามลำดับ ซึ่งเท่ากันแล้วแถวที่ t_4 กับ t_6 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 2 เท่ากันด้วย

นอกจากนี้จากตารางที่ 3.2 ยังแสดงให้เห็นอีกว่าแอททริบิวต์ A ซึ่งเป็นเซตย่อยของเซตแอททริบิวต์ AB สามารถบ่งชี้แอททริบิวต์ X ได้เพียงค่าเดียวเสมอเช่นกัน เนื่องจาก

แถวที่ t_1 กับ t_7 มีค่าเซตแอททริบิวต์ A เท่ากับ 0 เท่ากันแล้วแถวที่ t_1 กับ t_7 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 1 เท่ากันด้วย

แถวที่ t_2, t_4 และ t_6 มีค่าเซตแอททริบิวต์A เท่ากับ 2 แล้วแถวที่ t_2, t_4 และ t_6 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 2 เพียงค่าเดียว

แถวที่ t_3 กับ t_5 มีค่าเซตแอททริบิวต์ A เท่ากับ 2 เท่ากันแล้วแถวที่ t_3 กับ t_5 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 3 เท่ากันด้วย

ดังนั้น จะสรุปได้ว่า $AB \xrightarrow[P]{} X$ เป็นฟังก์ชันการขึ้นต่อกันแบบบางส่วน เนื่องจากมีเซตย่อย A ของแอททริบิวต์ AB ที่ $A \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกัน

ตารางที่ 3.2 ตารางฐานข้อมูลเชิงสัมพันธ์

Tuple	A	B	X
t_1	0	1	1
t_2	1	2	2
t_3	2	2	3
t_4	1	1	2
t_5	2	2	3
t_6	1	1	2
t_7	0	1	1

นิยามที่ 3.3 ฟังก์ชันการขึ้นต่อกันแบบเต็ม (Full Functional Dependency)
กำหนดให้ $A \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกัน จะกล่าวว่า แอททริบิวต์ A บ่งชี้แอททริบิวต์ X เป็นฟังก์ชันการขึ้นต่อกันแบบเต็ม เขียนแทนด้วย $A \xrightarrow[F]{} X$ ก็ต่อเมื่อ

$$\nexists B \subset A \; ; \; B \longrightarrow X$$

กำหนดให้ $A \xrightarrow[F]{} X$ จะกล่าวได้ว่า เซตแอททริบิวต์ A เป็นข้างซ้ายของฟังก์ชันขึ้นต่อกันแบบเต็ม ซึ่งเรียกว่า ดีเทอมิแนนท์แบบเต็ม (Full Determinant) และเซตแอททริบิวต์ X เป็นข้างขวาของฟังก์ชันขึ้นต่อกันแบบเต็ม ซึ่งเรียกว่า ออบเจคของดีเทอมิแนนท์แบบเต็ม

<u>ตัวอย่างที่ 3.4</u> เซตแอททริบิวต์ AB บ่งชี้แอททริบิวต์ X เป็นฟังก์ชันการขึ้นต่อกันแบบเต็ม

จากตารางที่ 3.3 แสดงให้เห็นว่าแต่ละค่าของเซตแอททริบิวต์ AB บ่งชี้ค่าแอททริบิวต์ X ได้เพียงค่าเดียวเสมอ เนื่องจาก

แถวที่ t_1 กับ t_7 มีค่าเซตแอททริบิวต์ A และ B เท่ากับ 0 และ 1 ตามลำดับ ซึ่งเท่ากันแล้วแถวที่ t_1 กับ t_7 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 1 เท่ากันด้วย

แถวที่ t_2 มีค่าเซตแอททริบิวต์ A และ B เท่ากับ 1 และ 2 ตามลำดับ แล้วแถวที่ t_2 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 1 เพียงค่าเดียว

แถวที่ t_3 กับ t_5 มีค่าเซตแอททริบิวต์ A และ B เท่ากับ 2 และ 2 ตามลำดับ ซึ่งเท่ากันแล้วแถวที่ t_3 กับ t_5 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 3 เท่ากันด้วย

แถวที่ t_4 กับ t_6 มีค่าเซตแอททริบิวต์ A และ B เท่ากับ 1 และ 1 ตามลำดับ ซึ่งเท่ากันแล้วแถวที่ t_4 กับ t_6 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 2 เท่ากันด้วย

นอกจากนี้จากตารางที่ 3.3 ยังแสดงให้เห็นอีกว่าเซตย่อยทั้งหมดของแอททริบิวต์ AB ไม่สามารถบ่งชี้แอททริบิวต์ X ได้ กล่าวคือ

แอททริบิวต์ A ซึ่งเป็นเซตย่อยของแอททริบิวต์ AB ไม่สามารถบ่งชี้แอททริบิวต์ X ได้ เนื่องจากแถวที่ t_2, t_4 และ t_6 มีค่าเซตแอททริบิวต์ A เท่ากับ 1 เท่ากัน แต่แถวที่ t_2 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 1 ส่วนแถวที่ t_4 กับ t_6 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 2 ดังนั้น แอททริบิวต์ A ไม่สามารถบ่งชี้แอททริวต์ X ได้ ($A \not\longrightarrow X$)

แอททริบิวต์ B ซึ่งเป็นเซตย่อยของแอททริบิวต์ AB ไม่สามารถบ่งชี้แอททริวต์ X ได้ เนื่องจากแถวที่ t_1, t_4, t_6 และ t_7 มีค่าเซตแอททริบิวต์ B เท่ากับ 1 เท่ากัน แต่แถวที่ t_1 กับ t_7 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 1 ส่วนแถวที่ t_4 กับ t_6 บ่งชี้ค่าแอททริบิวต์ X เท่ากับ 2 ดังนั้น แอททริบิวต์ B ไม่สามารถบ่งชี้แอททริวต์ X ได้ ($B \not\longrightarrow X$)

ดังนั้น จะสรุปได้ว่า $AB \xrightarrow[F]{} X$ เป็นฟังก์ชันการขึ้นต่อกันแบบเต็ม เนื่องจากไม่มีเซตย่อยทั้งหมดของแอททริบิวต์ AB ที่ เซตย่อยเหล่านั้นบ่งชี้แอททริบิวต์ X ได้

ตารางที่ 3.3 ตารางฐานข้อมูลเชิงสัมพันธ์

Tuple	A	B	X
t_1	0	1	1
t_2	1	2	1
t_3	2	2	3
t_4	1	1	2
t_5	2	2	3
t_6	1	1	2
t_7	0	1	1

3.2 คุณสมบัติของฟังก์ชันการขึ้นต่อกัน

ฟังก์ชันการขึ้นต่อกันมีคุณสมบัติตามสัจจพจน์ของอาร์มสตอง (Armstrong's Axiom) ดังนี้

1. กฎการสะท้อน (Reflexivity Rule)

   ถ้า $Y \subseteq X$ แล้ว $X \longrightarrow Y$

2. กฎการขยาย (Augmentation Rule)

   ถ้า $X \longrightarrow Y$ แล้ว $XZ \longrightarrow YZ$

3. กฎการถ่ายทอด (Transitive Rule)

   ถ้า $X \longrightarrow Y$ และ $Y \longrightarrow Z$ แล้ว $X \longrightarrow Z$

4. กฎการบ่งชี้ตัวเอง (Self Determination Rule)

   $A \longrightarrow A$

5. กฎการยูเนียนเพิ่ม (Additivity Union Rule)

   ถ้า $X \longrightarrow Y$ และ $X \longrightarrow Z$ แล้ว $X \longrightarrow YZ$

6. กฎการแตก (Projectivity Decomposition Rule)

   ถ้า $X \longrightarrow YZ$ แล้ว $X \longrightarrow Y$ และ $X \longrightarrow Z$

7. กฎการถ่ายทอดแบบเทียม (Pseudo Transitivity Rule)

   ถ้า $X \longrightarrow Y$ และ $WY \longrightarrow Z$ แล้ว $XW \longrightarrow Z$

3.3 การวิเคราะห์นิยามแอททริบิวต์รีดัก

3.3.1 ความสอดคล้องกันระหว่างพื้นที่ทางบวกกับฟังก์ชันการขึ้นต่อกัน

จากนิยามแอททริบิวต์รีดักบนทฤษฎีรัฟเซตในบทที่ 2 ได้กล่าวว่า แอททริบิวต์รีดักบนทฤษฎีรัฟเซตนั้นสามารถค้นหาได้จากการคำนวณพื้นที่ทางบวก $POS_R(D)$ และจากนิยามพื้นที่ทางบวกนั้นได้จากการคำนวณค่าประมาณขอบเขตล่าง

จากนิยามค่าประมาณขอบเขตล่างนั้นได้กล่าวว่า เป็นการยูเนียนสมาชิกทั้งหมดที่อยู่ใน U/IND(R) โดยขณะที่สมาชิกทั้งหมดเหล่านั้นจะต้องเป็นเซตย่อยของ U/IND(D) ด้วย

จากนิยาม U/IND(R) ได้กล่าวว่า U/IND(R) เป็นเซตของคลาสสมมูลทั้งหมดของแอททริบิวต์ R กล่าวคือ สมาชิกของ U/IND(R) เป็นเซตของคลาสสมมูลกันทั้งหมดของแอททริบิวต์ R ซึ่งคลาสสมมูลกันเหล่านั้นจะถูกแบ่งตามค่าโดเมนทั้งหมดของแอททริบิวต์ R นั่นคือ คลาสสมมูลกันของแอททริบิวต์ R หมายถึงแถวข้อมูลที่มีค่าโดเมนของแอททริบิวต์ R ทั้งหมดเท่ากันจะถูกจัดอยู่ในคลาสเดียวกัน หรือกล่าวอีกนัยหนึ่งว่า สมาชิกแต่ละตัวที่อยู่ใน U/IND(R) เป็นคลาสสมมูลกันของแอททริบิวต์ R ดังนั้น จำนวนสมาชิกที่อยู่ใน U/IND(R) จะเท่ากับจำนวนค่าโดเมนของแอททริบิวต์ R ทั้งหมด ซึ่งคลาสสมมูลกันของแอททริบิวต์ R หมายถึงแถวข้อมูลที่มีค่าโดเมนของแอททริบิวต์ R ทั้งหมดเท่ากันจะถูกจัดอยู่ในคลาสเดียวกัน

เช่นเดียวกันกับ U/IND(D) นั้นเป็นเซตของคลาสสมมูลกันทั้งหมดของแอททริบิวต์ตัดสินใจ D กล่าวคือ สมาชิกของ U/IND(D) เป็นเซตของคลาสสมมูลกันทั้งหมดของแอททริบิวต์ตัดสินใจ D ซึ่งคลาสสมมูลกันเหล่านั้นจะถูกแบ่งตามค่าโดเมนทั้งหมดของแอททริบิวต์ตัดสินใจ D นั่นคือ คลาสสมมูลกันของแอททริบิวต์ตัดสินใจ D หมายถึงแถวข้อมูลที่มีค่าโดเมนของแอททริบิวต์ตัดสินใจ D ทั้งหมดเท่ากันจะถูกจัดอยู่ในคลาสตัดสินใจคลาสเดียวกัน หรือกล่าวอีกนัยหนึ่งว่าสมาชิกแต่ละตัวที่อยู่ใน U/IND(D) เป็นคลาสสมมูลกันของแอททริบิวต์ตัดสินใจ D ดังนั้น จำนวนสมาชิกที่อยู่ใน U/IND(D) จะเท่ากับจำนวนค่าโดเมนของแอททริบิวต์ตัดสินใจ D ทั้งหมด ซึ่งคลาสสมมูลกันของแอททริบิวต์ตัดสินใจ D หมายถึงแถวข้อมูลที่มีค่าโดเมนของแอททริบิวต์ตัดสินใจ D ทั้งหมดเท่ากันจะถูกจัดอยู่ในคลาสตัดสินใจเดียวกัน

จากนิยามข้างต้นดังกล่าว ทำให้ทราบว่า พื้นที่ทางบวกเป็นเซตของแถวข้อมูลทั้งหมดที่เป็นสมาชิกของ U/IND(R) ซึ่งแถวข้อมูลทั้งหมดเหล่านั้นจะเป็นเซตย่อยของแถวข้อมูลทั้งหมดที่เป็นสมาชิกของ U/IND(D) ด้วย ดังนั้น พื้นที่ทางบวกจะเป็นเซตของแถวข้อมูลทั้งหมดที่มีค่าแอททริบิวต์ R เท่ากันแล้วแถวข้อมูลทั้งหมดเหล่านั้นจะเป็นแถวข้อมูลที่มีค่าแอททริบิวต์ตัดสินใจเท่ากันด้วย ด้วยเหตุนี้จึงกล่าวได้ว่าแถวข้อมูลที่ได้รับจากนิยามพื้นที่ทางบวกจะสอดคล้องกับนิยามฟังก์ชันการขึ้นต่อกัน โดยจะกำหนดให้ข้างซ้ายของฟังก์ชันการขึ้นต่อกันเป็นเซตแอททริบิวต์ R และข้างขวาของฟังก์ชันการขึ้นต่อกันเป็นเซตแอททริบิวต์ตัดสินใจ D ดังนั้น จากความสอดคล้อง

กันดังกล่าว เราจึงได้นำทฤษฎีฟังก์ชันการขึ้นต่อกันมาช่วยในการคำนวณหาแอททริบิวต์รีดักแทนที่การคำนวณจากพื้นที่ทางบวก

ตัวอย่างที่ 3.5 แถวข้อมูลที่อยู่ใน $POS_R(D)$ และแถวข้อมูลที่ไม่อยู่ใน $POS_R(D)$

จากตารางที่ 3.4 C = {Weight, Door, Size, Cylinder} และ D = {Mileage}

สมมติให้ R = {Weight, Size}

ตารางที่ 3.4 แสดงตารางตัดสินใจของข้อมูลรถเพื่อทำนายระยะทาง ที่มีแอททริบิวต์ Weight, Door Size และ Cylinder เป็นแอททริบิวต์เงื่อนไข และแอททริบิวต์ Mileage เป็นแอททริบิวต์ตัดสินใจ

U	Weight	Door	Size	Cylinder	Mileage
x_1	Low	2	Compact	4	High
x_2	Low	4	Sub	6	Low
x_3	Med	4	Compact	4	High
x_4	High	2	Compact	6	Low
x_5	High	4	Compact	4	Low
x_6	Low	4	Compact	4	High
x_7	High	4	Sub	6	Low
x_8	Low	2	Sub	6	Low
x_9	High	2	Sub	4	High
x_{10}	High	2	Sub	4	Low

U/IND(R) เป็นเซตของคลาสสมมูลทั้งหมดของแอททริบิวต์ R ซึ่งคลาสสมมูลของแอททริบิวต์ R เป็นเซตของแถวข้อมูลที่มีค่าโดเมนแอททริบิวต์ R ทั้งหมดเท่ากัน ดังนั้นจำนวนคลาสสมมูลของแอททริบิวต์ R ขึ้นอยู่กับค่าโดเมนของแอททริบิวต์ R ทั้งหมด

U/IND(R) = $\{\{x_1, x_6\}, \{x_2, x_8\}, \{x_3\}, \{x_4, x_5\}, \{x_7, x_9, x_{10}\}\}$ แสดงดังตารางที่ 3.5

U/IND(D) เป็นเซตของคลาสสมมูลทั้งหมดของแอททริบิวต์ตัดสินใจ D ซึ่งคลาสสมมูลของแอททริบิวต์ตัดสินใจ D เป็นเซตของแถวข้อมูลที่มีค่าโดเมนแอททริบิวต์ตัดสินใจ D เท่ากัน ดังนั้น จำนวนคลาสสมมูลกันของแอททริบิวต์ตัดสินใจ D ขึ้นอยู่กับค่าโดเมนของแอททริบิวต์ตัดสินใจ D ทั้งหมด

$U/IND(D) = \{\{x_1, x_3, x_6, x_9\}, \{x_2, x_4, x_5, x_7, x_8, x_{10}\}\}$ แสดงดังตารางที่ 3.5

กำหนดให้ $X_1 = U/IND[Mileage = High] = \{x_1, x_3, x_6, x_9\}$

และ $X_2 = U/IND[Mileage = Low] = \{x_2, x_4, x_5, x_7, x_8, x_{10}\}$

$$\underline{RX_1} = \bigcup \{U/IND(R) : U/IND(R) \subseteq U/IND[Mileage = High]\}$$

$$\underline{RX_1} = (\{x_1, x_6\} \subseteq \{x_1, x_3, x_6, x_9\}) \cup (\{x_2, x_8\} \subseteq \{x_1, x_3, x_6, x_9\}) \cup (\{x_3\} \subseteq \{x_1, x_3, x_6, x_9\})$$
$$\cup (\{x_4, x_5\} \subseteq \{x_1, x_3, x_6, x_9\}) \cup (\{x_7, x_9, x_{10}\} \subseteq \{x_1, x_3, x_6, x_9\})$$

ดังนั้น $\underline{RX_1} = \{x_1, x_6, x_3\}$

$$\underline{RX_2} = \bigcup \{U/IND(R) : U/IND(R) \subseteq U/IND[Mileage = Low]\}$$

$$\underline{RX_2} = (\{x_1, x_6\} \subseteq \{x_2, x_4, x_5, x_7, x_8, x_{10}\}) \cup (\{x_2, x_8\} \subseteq \{x_2, x_4, x_5, x_7, x_8, x_{10}\})$$
$$\cup (\{x_3\} \subseteq \{x_2, x_4, x_5, x_7, x_8, x_{10}\}) \cup (\{x_4, x_5\} \subseteq \{x_2, x_4, x_5, x_7, x_8, x_{10}\})$$
$$\cup (\{x_7, x_9, x_{10}\} \subseteq \{x_2, x_4, x_5, x_7, x_8, x_{10}\})$$

ดังนั้น $\underline{RX_2} = \{x_2, x_8, x_4, x_5, x_7\}$

$$POS_R(D) = \bigcup_{X \in U/IND(D)} \underline{RX} = \underline{RX_1} \cup \underline{RX_2} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$$

จากตัวอย่างแสดงให้เห็นว่าแถวทั้งหมดที่อยู่ในพื้นที่ทางบวก $POS_R(D)$ เป็นแถวข้อมูลที่มีคุณสมบัติเป็นฟังก์ชันการขึ้นต่อกัน เนื่องจากแถวทั้งหมดที่อยู่ในพื้นที่ทางบวก $POS_R(D)$ เป็นแถวข้อมูลที่เมื่อมีค่าแอททริบิวต์ R เท่ากันแล้วแถวข้อมูลเหล่านั้นจะมีค่าแอททริบิวต์ D เท่ากันด้วย หรือกล่าวอีกนัยนึงว่า ถ้าแถวข้อมูลใด ๆ มีค่าแอททริบิวต์ D เท่ากันแล้วจะต้องไม่มีแถวข้อมูลอื่นที่มีค่าแอททริบิวต์ R เท่ากับแถวข้อมูลเหล่านี้ ยกตัวอย่างเช่น แถวข้อมูล x_1, x_6, x_3 ที่มีค่าแอททริบิวต์ D เท่ากับ High เท่ากัน แต่ไม่มีแถวข้อมูลอื่นใดในตารางที่ปรากฏว่ามีค่าแอททริบิวต์ R เท่ากันกับแถวข้อมูล x_1, x_6, x_3 ทั้ง 3 แถวนี้เลย ดังนั้น จากตัวอย่างแสดงให้เห็นว่าแถวทั้งหมดที่อยู่ในพื้นที่ทางบวก $POS_R(D)$ เป็นแถวข้อมูลที่มีค่าข้อมูลสอดคล้องกันระหว่างเซตแอททริบิวต์ R กับแอททริบิวต์ตัดสินใจ D นั่นคือ แถวที่ x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 มีค่าข้อมูลสอดคล้องกันระหว่างเซตแอททริบิวต์ R กับแอททริบิวต์ตัดสินใจ D กล่าวคือ ถ้าแถวข้อมูลใดมีค่าแอททริบิวต์ R เท่ากันแล้วแถวข้อมูลจะมีค่าแอททริบิวต์ตัดสินใจ D เท่ากันด้วย

จากตัวอย่างจะกล่าวได้ว่าแถวที่ไม่อยู่ในพื้นที่ทางบวก $POS_R(D)$ เป็นแถวข้อมูลที่ก่อให้เกิดค่าข้อมูลไม่สอดคล้องกันระหว่างเซตแอททริบิวต์ R กับแอททริบิวต์ D หรือกล่าวอีกนัยนึงว่าถ้าแถวข้อมูลใดมีค่าแอททริบิวต์ R เท่ากันแล้วแถวข้อมูลเหล่านั้นมีค่าแอททริบิวต์ตัดสินใจ D หลายค่าซึ่งเรียกเหตุการณ์นี้ว่าวันทูเมนี่ ยกตัวอย่างเช่น แถวข้อมูล x_9 กับ x_{10} มีค่าแอททริบิวต์ R เท่ากับ High, Sub ซึ่งเท่ากัน แต่แถวข้อมูล x_9 มีค่าแอททริบิวต์ตัดสินใจ เท่ากับ High ส่วนแถวข้อมูล x_{10} มีค่าแอททริบิวต์ตัดสินใจเท่ากับ Low ทำให้เกิดค่าข้อมูลไม่สอดคล้องกัน ดังนั้นแถวข้อมูล x_9 กับ x_{10} จะไม่อยู่ในพื้นที่ทางบวก $POS_R(D)$ ซึ่งจะสอดคล้องกับการคำนวณหา $POS_R(D)$

ดังนั้น ถ้า $POS_R(D) = U$ แสดงว่า $R \longrightarrow D$ เป็นฟังก์ชันการขึ้นต่อกัน กล่าวคือ ถ้าแถวข้อมูลใด ๆ ใน U มีค่าแอททริบิวต์ R เท่ากันแล้วแถวข้อมูลเหล่านั้นจะมีค่าแอททริบิวต์ตัดสินใจ D เท่ากันด้วย ส่วนถ้า $POS_R(D) \neq U$ แสดงว่า $R \longrightarrow D$ ไม่เป็นฟังก์ชันการขึ้นต่อกัน เขียนแทนด้วย $R \not\longrightarrow D$ กล่าวคือ มีบางแถวข้อมูลใน U มีค่าแอททริบิวต์ R เท่ากันแต่มีค่าแอททริบิวต์ตัดสินใจ D ไม่เท่ากันซึ่งแถวข้อมูลเหล่านั้นจะไม่อยู่ใน$POS_R(D)$

จากตัวอย่างจะสรุปได้ว่าแถวข้อมูลที่ได้รับจากพื้นที่ทางบวกสอดคล้องกับคุณสมบัติฟังก์ชันการขึ้นต่อกัน ($POS_R(D) \equiv R \longrightarrow D$) ดังนั้นเราสามารถนำฟังก์ชันการขึ้นต่อกันมาช่วยในการหาแอททริบิวต์รีดักแทนที่การคำนวณหาจากพื้นที่ทางบวก

ตารางที่ 3.5 แสดงตารางตัดสินใจของข้อมูลรถเพื่อทำนายระยะทาง ที่มีแอททริบิวต์ Weight และ Size เป็นแอททริบิวต์เงื่อนไข และแอททริบิวต์ Mileage เป็นแอททริบิวต์ตัดสินใจ

U	Weight	Size	Mileage
x_1	Low	Compact	High
x_2	Low	Sub	Low
x_3	Med	Compact	High
x_4	High	Compact	Low
x_5	High	Compact	Low
x_6	Low	Compact	High
x_7	High	Sub	Low
x_8	Low	Sub	Low
x_9	High	Sub	High
x_{10}	High	Sub	Low

3.3.2 ความสอดคล้องกันระหว่างนิยามแอททริบิวต์รีดักบนรัฟเซตกับฟังก์ชันการขึ้นต่อกันแบบเต็ม

จากนิยามแอททริบิวต์รีดักบนทฤษฎีรัฟเซตในบทที่ 2 ได้กล่าวว่า R' เป็นแอททริบิวต์รีดักในตารางตัดสินใจ ถ้า $POS_{R'}(D) = U$ และ $\forall B \subset R'; POS_B(D) \neq U$ โดยที่ $R' \subseteq C$

จากความสอดคล้องกันระหว่างพื้นที่ทางบวกกับฟังก์ชันการขึ้นต่อกันในหัวข้อที่ 3.3.1 แสดงให้เห็นว่า $POS_{R'}(D) = U$ หมายความว่า $R' \longrightarrow D$ เป็นฟังก์ชันการขึ้นต่อกันและ $POS_B(D) \neq U$ หมายความว่า $B \longrightarrow D$ ไม่เป็นฟังก์ชันการขึ้นต่อกัน $(B \not\longrightarrow D)$ ดังนั้น จากนิยามแอททริบิวต์รีดัก จะกล่าวได้ว่า R' เป็นแอททริบิวต์รีดักในตารางตัดสินใจ ถ้า $R' \longrightarrow D$ เป็นฟังก์ชันการขึ้นต่อกัน และ $\forall B \subset R'; B \not\longrightarrow D$ หรือกล่าวได้ว่า R' เป็นแอททริบิวต์รีดักในตารางตัดสินใจ ถ้า $R' \xrightarrow[F]{} D$ เป็นฟังก์ชันการขึ้นต่อกันแบบเต็ม ซึ่งจะเรียก R' ได้ว่าเป็นดีเทอร์มิแนนท์แบบเต็ม

<u>ตัวอย่างที่ 3.6</u> R' เป็นแอททริบิวต์รีดักบนทฤษฎีรัฟเซตเทียบเท่ากับ R' เป็นดีเทอร์มิแนนท์แบบเต็มบนฟังก์ชันการขึ้นต่อกัน

จากข้อมูลในตารางที่ 3.3 กำหนดให้ C = {AB} และ D = {X} เมื่อทำการทำคำนวณพื้นที่ทางบวกของเซตแอททริบิวต์รีดักที่เป็นไปได้ทั้งหมดในตารางจะได้

$POS_A(D) = \{x_1, x_3, x_5, x_7\} \neq U$ แสดงว่า $A \not\longrightarrow D$

$POS_B(D) = \{\} \neq U$ แสดงว่า $B \not\longrightarrow D$

$POS_{AB}(D) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$ แสดงว่า $AB \longrightarrow D$

จากตัวอย่างจะกล่าวได้ว่า R' = AB เป็นแอททริบิวต์รีดักบนทฤษฎีรัฟเซต เนื่องจาก $POS_{AB}(D) = U$ และ $POS_A(D) \neq U$ กับ $POS_B(D) \neq U$

จากตัวอย่างจะกล่าวได้ว่า R' เป็นดีเทอร์มิแนนท์แบบเต็มของฟังก์ชันการขึ้นต่อกันแบบต็ม เนื่องจาก $AB \longrightarrow D$ และ $A \not\longrightarrow D$ กับ $B \not\longrightarrow D$

ดังนั้น จะสรุปได้ว่า R' เป็นแอททริบิวต์รีดักบนทฤษฎีรัฟเซตเทียบเท่ากับ R' เป็นดีเทอร์มิแนนท์แบบเต็มบนทฤษฎีฟังก์ชันการขึ้นต่อกัน

งานวิจัยนี้จึงได้นำทฤษฎีฟังก์ชันการขึ้นต่อกันมาช่วยในการค้นหาแอททริบิวต์รีดักแทนที่การคำนวณจากวิธีพื้นที่ทางบวก โดยการค้นหาแอททริบิวต์รีดักบนทฤษฎีรัฟเซตเทียบเท่ากับการค้นหาดีเทอร์มิแนนแบบเต็มบนฟังก์ชันการขึ้นต่อกัน เมื่อกำหนดให้แอททริบิวต์ตัดสินใจเป็นออบเจคของดีเทอร์มิแนนท์แบบเต็ม

3.4 นิยามแอททริบิวต์รีดักบนทฤษฎีฟังก์ชันการขึ้นต่อกัน

นิยามที่ 3.4 ฟังก์ชันการขึ้นต่อกันระหว่างแอททริบิวต์ R' และแอททริบิวต์ตัดสินใจ D กำหนดให้ $T(U, A = C \cup D, V, f)$ เป็นตารางตัดสินใจ และ $R' \subseteq C$ จะกล่าวว่า ฟังก์ชันการขึ้นต่อกันระหว่างแอททริบิวต์ R' และแอททริบิวต์ตัดสินใจ D หรือเรียกอีกอย่างหนึ่งว่า แอททริบิวต์ R' บ่งชี้แอททริบิวต์ตัดสินใจ D เขียนแทนด้วย $R' \longrightarrow D$ ก็ต่อเมื่อ แต่ละค่าของแอททริบิวต์ R' สามารถบ่งชี้ค่าแอททริบิวต์ตัดสินใจ D ได้เพียงค่าเดียวเสมอ เขียนแทนด้วย

$$\forall(x_i, x_j): \text{ถ้า } x_i(R') = x_j(R') \text{ แล้ว } x_i(D) = x_j(D) \text{ โดยที่ } x_i, x_j \in U$$

ตัวอย่างที่ 3.6 ฟังก์ชันการขึ้นต่อกันระหว่างแอททริบิวต์ R' และแอททริบิวต์ตัดสินใจ D จากข้อมูลในตารางที่ 3.6

ตารางที่ 3.6 แสดงตารางตัดสินใจของข้อมูลรถเพื่อทำนายระยะทาง ที่มีแอททริบิวต์ Weight, Door, Size และ Cylinder เป็นแอททริบิวต์เงื่อนไข และแอททริบิวต์ Mileage เป็นแอททริบิวต์ตัดสินใจ

U	Weight	Door	Size	Cylinder	Mileage
x_1	Low	2	Compact	4	High
x_2	Low	4	Sub	6	Low
x_3	Med	4	Compact	4	High
x_4	High	2	Compact	6	Low
x_5	High	4	Compact	4	Low
x_6	Low	4	Compact	4	High
x_7	High	4	Sub	6	Low
x_8	Low	2	Sub	6	Low

กำหนดให้ R' = {Weight, Size} และ D = {Mileage}

จากตารางที่ 3.7 แสดงให้เห็นว่าแถวข้อมูลมีค่าแอททริบิวต์ Weight และ Size เท่ากันแล้ว สามารถบ่งชี้ค่าโดเมนของแอททริบิวต์ Mileage ค่าเดียวเสมอ นั่นคือ

แถวข้อมูล x_1 และ x_6 มีค่าแอททริบิวต์ Weight และ Size เท่ากับ Low และ Compact ตามลำดับ ซึ่งเท่ากันแล้วแถวข้อมูล x_1 และ x_6 ยังมีค่าแอททริบิว Mileage เท่ากับ high เท่ากันด้วย

แถวข้อมูล x_2 และ x_8 มีค่าแอททริบิวต์ Weight และ Size เท่ากับ Low และ Sub ตามลำดับ ซึ่งเท่ากันแล้วแถวข้อมูล x_2 และ x_8 ยังมีค่าแอททริบิวต์ Mileage เท่ากับ Low เท่ากันด้วย

แถวข้อมูล x_3 มีค่าแอททริบิวต์ Weight และ Size เท่ากับ Med และ Compact ตามลำดับ ซึ่งไม่เท่ากับแถวข้อมูลอื่น ๆ ในตารางและแถวข้อมูล x_3 มีค่าแอททริบิวต์ Mileage เท่ากับ High เพียงค่าเดียว

แถวข้อมูล x_4 และ x_5 มีค่าแอททริบิวต์ Weight และ Size เท่ากับ High และ Compact ตามลำดับ ซึ่งเท่ากันแล้วแถวข้อมูล x_4 และ x_5 ยังมีค่าแอททริบิวต์ Mileage เท่ากับ Low เท่ากันด้วย

แถวข้อมูล x_7 มีค่าแอททริบิวต์ Weight และ Size เท่ากับ High และ Sub ตามลำดับ ซึ่งไม่เท่ากับแถวข้อมูลอื่น ๆ ในตารางและแถวข้อมูล x_7 มีค่าแอททริบิวต์ Mileage เท่ากับ High

ตารางที่ 3.7 แสดงฟังก์ชันการขึ้นต่อกันของข้อมูลรถระหว่างแอททริบิวต์ Weight, Size กับแอททริบิวต์ Mileage

U	Weight	Size	Mileage
x_1 x_6	Low	Compact	High
x_2 x_8	Low	Sub	Low
x_3	Med	Compact	High
x_4 x_5	High	Compact	Low
x_7	High	Sub	Low

นิยามที่ 3.5 ฟังก์ชันการขึ้นต่อกันบางส่วนระหว่างแอททริบิวต์ R' กับแอททริบิวต์ตัดสินใจ D กำหนดให้ $T(U, A = C \cup D, V, f)$ เป็นตารางตัดสินใจ, $R' \subseteq C$ และ $R' \longrightarrow D$ เป็นฟังก์ชันการขึ้นต่อกันระหว่างแอททริบิวต์ R' กับแอททริบิวต์ตัดสินใจ D จะกล่าวว่า $R' \xrightarrow[P]{} D$ เป็นฟังก์ชันการขึ้นต่อกันบางส่วนระหว่างแอททริบิวต์ R' และแอททริบิวต์ตัดสินใจ D ก็ต่อเมื่อ

$$\exists B \subset R';\ B \longrightarrow D \text{ โดยที่ } B \neq \phi$$

ตัวอย่างที่ 3.7 ฟังก์ชันการขึ้นต่อกันบางส่วนระหว่างแอททริบิวต์ R' และแอททริบิวต์ตัดสินใจ D จากข้อมูลในตารางที่ 3.6

กำหนดให้ R' = {Weight, Size, Cylinder} และ D = {Mileage}

จากตารางที่ 3.8 แสดงให้เห็นว่าแถวข้อมูลมีค่าแอททริบิวต์ Weight, Size และ Cylinder เท่ากันแล้วสามารถบ่งชี้ค่าโดเมนของแอททริบิวต์ Mileage ค่าเดียวเสมอ นั่นคือ

แถวข้อมูล x_1 และ x_6 มีค่าแอททริบิวต์ Weight, Size และ Cylinder เท่ากับ Low, Compact และ 4 ตามลำดับ ซึ่งเท่ากันแล้วยังมีค่าแอททริบิวต์ Mileage เท่ากับ High เท่ากันเพียงค่าเดียวด้วย

แถวข้อมูล x_2 และ x_8 มีค่าแอททริบิวต์ Weight, Size และ Cylinder เท่ากับ Low, Sub และ 6 ตามลำดับ ซึ่งเท่ากันแล้วยังมีค่าแอททริบิวต์ Mileage เท่ากับ Low เท่ากันเพียงค่าเดียวด้วย

แถวข้อมูล x_3 มีค่าแอททริบิวต์ Weight, Size และ Cylinder เท่ากับ Med, Compact และ 4 ตามลำดับ ซึ่งไม่เท่ากับแถวข้อมูลอื่น ๆ ในตารางและแถวข้อมูล x_3 มีค่าแอททริบิวต์ Mileage เท่ากับ High เพียงค่าเดียว

แถวข้อมูล x_4 มีค่าแอททริบิวต์ Weight, Size และ Cylinder เท่ากับ High, Compact และ 6 ตามลำดับ ซึ่งไม่เท่ากับแถวข้อมูลอื่น ๆ ในตารางและแถวข้อมูล x_4 มีค่าแอททริบิวต์ Mileage เท่ากับ Low เพียงค่าเดียว

แถวข้อมูล x_5 มีค่าแอททริบิวต์ Weight, Size และ Cylinder เท่ากับ High, Compact และ 4 ตามลำดับ ซึ่งไม่เท่ากับแถวข้อมูลอื่น ๆ ในตารางและ แถวข้อมูล x_5 มีค่าแอททริบิวต์ Mileage เท่ากับ Low เพียงค่าเดียว

แถวข้อมูล x_7 มีค่าแอททริบิวต์ Weight, Size และ Cylinder เท่ากับ High, Sub และ 6 ตามลำดับ ซึ่งไม่เท่ากับแถวข้อมูลอื่น ๆ ในตารางและ แถวข้อมูล x_7 มีค่าแอททริบิวต์ Mileage เท่ากับ Low เพียงค่าเดียว

ตารางที่ 3.8 แสดงฟังก์ชันการขึ้นต่อกันของข้อมูลรถระหว่างแอททริบิวต์ Weight, Size และ Cylinder กับแอททริบิวต์ Mileage

U	Weight	Size	Cylinder	Mileage
x_1 x_6	Low	Compact	4	High
x_2 x_8	Low	Sub	6	Low
x_3	Med	Compact	4	High
x_4	High	Compact	6	Low
x_5	High	Compact	4	Low
x_7	High	Sub	6	Low

สมมติให้ B = {Weight, Cylinder} ซึ่งเป็นเซตย่อยของแอททริบิวต์ R' = {Weight, Size, Cylinder}

จากตารางที่ 3.9 แสดงให้เห็นว่าแถวข้อมูลมีค่าแอททริบิวต์ Weight และ Cylinder เท่ากัน แล้วสามารถบ่งชี้ค่าโดเมนของแอททริบิวต์ Mileage ค่าเดียวเสมอ นั่นคือ

แถวข้อมูล x_1 และ x_6 มีค่าแอททริบิวต์ Weight และ Cylinder เท่ากับ Low และ 4 ตามลำดับ ซึ่งเท่ากันแล้วแถวข้อมูล x_1 และ x_6 ยังมีค่าแอททริบิวต์ Mileage เท่ากับ High เท่ากันเพียงค่าเดียวด้วย

แถวข้อมูล x_2 และ x_8 มีค่าแอททริบิวต์ Weight และ Cylinder เท่ากับ Low และ 6 ตามลำดับ ซึ่งเท่ากันแล้วแถวข้อมูล x_2 และ x_8 ยังมีค่าแอททริบิวต์ Mileage เท่ากับ Low เท่ากันเพียงค่าเดียวด้วย

แถวข้อมูล x_3 มีค่าแอททริบิวต์ Weight และ Cylinder เท่ากับ Med และ 4 ตามลำดับ ซึ่งไม่เท่ากับแถวข้อมูลอื่น ๆ ในตารางและแถวข้อมูล x_3 มีค่าแอททริบิวต์ Mileage เท่ากับ High เพียงค่าเดียว

แถวข้อมูล x_4 และ x_7 มีค่าแอททริบิวต์ Weight และ Cylinder เท่ากับ High และ 6 ตามลำดับ ซึ่งเท่ากันแล้วแถวข้อมูล x_4 และ x_7 ยังมีค่าแอททริบิวต์ Mileage เท่ากับ Low เท่ากันเพียงค่าเดียวด้วย

แถวข้อมูล x_5 มีค่าแอททริบิวต์ Weight และ Cylinder เท่ากับ High และ 4 ตามลำดับซึ่งไม่เท่ากับแถวข้อมูลอื่น ๆ ในตารางและแถวข้อมูล x_5 มีค่าแอททริบิวต์ Mileage เท่ากับ Low เพียงค่าเดียว

แสดงว่ามีเซตย่อย B = {Weight, Cylinder} ของแอททริบิวต์ R' = {Weight, Size, Cylinder} โดยที่ $B \longrightarrow D$ จริง ดังนั้น $R' \xrightarrow[P]{} D$ เป็นฟังก์ชันการขึ้นต่อกันบางส่วน

ตารางที่ 3.9 แสดงฟังก์ชันการขึ้นต่อกันของข้อมูลรถระหว่างแอททริบิวต์ Weight และ Cylinder กับแอททริบิวต์ Mileage

U	Weight	Cylinder	Mileage
x_1 x_6	Low	4	High
x_2 x_8	Low	6	Low
x_3	Med	4	High
x_4 x_7	High	6	Low
x_5	High	4	Low

นิยามที่ 3.6 ฟังก์ชันการขึ้นต่อกันแบบเต็มระหว่างแอททริบิวต์ R' กับแอททริบิวต์ตัดสินใจ D กำหนดให้ $T(U, A = C \cup D, V, f)$ เป็นตารางตัดสินใจ, $R' \subseteq C$ และ $R' \longrightarrow D$ เป็นฟังก์ชันการขึ้นต่อกันระหว่างแอททริบิวต์ R' กับแอททริบิวต์ตัดสินใจ D จะกล่าวว่า $R' \xrightarrow[F]{} D$ เป็นฟังก์ชันการขึ้นต่อกันแบบเต็มระหว่างแอททริบิวต์ R' และแอททริบิวต์ตัดสินใจ D ก็ต่อเมื่อ

$$\nexists B \subset R';\ B \longrightarrow D \text{ โดยที่ } B \neq \phi$$

<u>ตัวอย่างที่ 3.8</u> ฟังก์ชันการขึ้นต่อกันแบบเต็มระหว่างแอททริบิวต์ R' และแอททริบิวต์ตัดสินใจ D จากข้อมูลในตารางที่ 3.6

กำหนดให้ R' = {Weight, Size} จากตัวอย่างที่ 3.6 พบว่าแถวข้อมูลมีค่าแอททริบิวต์ Weight และ Size เท่ากันแล้วสามารถบ่งชี้ค่าโดเมนของแอททริบิวต์ Mileage ค่าเดียวเสมอ แสดงว่า $R' \longrightarrow D$ ดังนั้น เซตย่อยทั้งหมดของแอททริบิวต์ R' = {Weight, Size} คือ {Weight} กับ {Size}

จากตารางที่ 3.10 แสดงให้เห็นว่าแถวข้อมูลมีค่าแอททริบิวต์ Weight เท่ากันแล้วไม่สามารถบ่งชี้ค่าโดเมนของแอททริบิวต์ Mileage ค่าเดียวเสมอ นั่นคือ แถวข้อมูล x_1, x_6 กับแถวข้อมูล x_2, x_8 มีค่าแอททริบิวต์ Weight เท่ากับ Low เท่ากัน แต่แถวข้อมูล x_1, x_6 มีค่าแอททริบิวต์ Mileage เท่ากับ High และแถวข้อมูล x_2, x_8 มีค่าแอททริบิวต์ Mileage เท่ากับ Low แสดงว่าแถวข้อมูลใด ๆ ที่มีค่าแอททริบิวต์ Weight เท่ากันแล้วไม่สามารถบ่งชี้แอททริบิวต์ Mileage เพียงค่าเดียวเสมอ แสดงว่าแอททริบิวต์ Weight ไม่สามารถบ่งชี้แอททริบิวต์ Mileage

ตารางที่ 3.10 แสดงฟังก์ชันการขึ้นต่อกันของข้อมูลรถระหว่างแอททริบิวต์ Weight กับแอททริบิวต์Mileage

U	Weight	Mileage
x_1 x_6	Low	High
x_2 x_8	Low	Low
x_3	Med	High
x_4 x_5 x_7	High	Low

จากตารางที่ 3.11 แสดงให้เห็นว่าแถวข้อมูลมีค่าแอททริบิวต์ Size เท่ากันแล้วไม่สามารถบ่งชี้ค่าโดเมนของแอททริบิวต์ Mileage ค่าเดียวเสมอ นั่นคือ แถวข้อมูล x_1, x_3, x_6 กับแถวข้อมูล x_4, x_5 มีค่าแอททริบิวต์ Size เท่ากับ Compact เท่ากัน แต่แถวข้อมูล x_1, x_3, x_6 มีค่าแอททริบิวต์ Mileage เท่ากับ High และแถวข้อมูล x_4, x_5 มีค่าแอททริบิวต์ Mileage เท่ากับ Low แสดงว่าแถวข้อมูลใด ๆ ที่มีค่าแอททริบิวต์ Size เท่ากันแล้วไม่สามารถบ่งชี้แอททริบิวต์ Mileage เพียงค่าเดียวเสมอ แสดงว่าแอททริบิวต์ Size ไม่สามารถบ่งชี้แอททริบิวต์ Mileage

ดังนั้น เซตย่อยทั้งหมด B คือ {Weight} และ {Size} ของแอททริบิวต์ R' = {Weight, Size} ซึ่งเซตย่อยทั้งหมดเหล่านั้นไม่สามารถบ่งชี้แอททริบิวต์ Mileage ได้ แสดงให้เห็นว่า $R' \xrightarrow[F]{} D$ เป็นฟังก์ชันการขึ้นต่อกันแบบเต็ม

ตารางที่ 3.11 แสดงฟังก์ชันการขึ้นต่อกันของข้อมูลรถระหว่างแอททริบิวต์ Size กับแอททริบิวต์ Mileage

U	Size	Mileage
x_1 x_3 x_6	Compact	High
x_2 x_7 x_8	Sub	Low
x_4 x_5	Compact	Low

นิยามที่ 3.7 แอททริบิวต์รีดักบนฟังก์ชันการขึ้นต่อกัน (Reduct Based on Functional Dependency) กำหนดให้ $T(U, A = C \cup D, V, f)$ เป็นตารางตัดสินใจ และ $R' \longrightarrow D$ เป็นฟังก์ชันการขึ้นต่อกันระหว่างแอททริบิวต์รีดัก R' กับแอททริบิวต์ตัดสินใจ D จะกล่าวว่า R' เป็นแอททริบิวต์รีดักของตารางตัดสินใจ ก็ต่อเมื่อ

$$R' \xrightarrow[F]{} D \text{ เป็นฟังก์ชันการขึ้นต่อกันแบบเต็ม โดยที่ } R' \subseteq C \text{ และ } R' \neq \phi$$

กำหนดให้ $R' \xrightarrow[F]{} D$ จะกล่าวได้ว่า เซตแอททริบิวต์รีดัก R' เป็นข้างซ้ายของฟังก์ชันการขึ้นต่อกันแบบเต็ม ซึ่งเรียกว่า ดีเทอมิแนนท์แบบเต็ม (Full Determinant) และเซตแอททริบิวต์ตัดสินใจ D เป็นข้างขวาของฟังก์ชันขึ้นต่อกันแบบเต็ม ซึ่งเรียกว่า ออบเจคของดีเทอมิแนนท์แบบเต็ม

<u>ตัวอย่างที่ 3.9</u> ตัวอย่างแอททริบิวต์รีดักบนฟังก์ชันการขึ้นต่อกัน

จากตารางที่ 3.6 C = {Weight, Door, Size, Cylinder} และ D = {Mileage} จากตัวอย่างที่ 3.6 กำหนดให้ R' = {Weight, Size} แสดงให้เห็นว่า $\{Weight, Size\} \longrightarrow \{Mileage\}$ เป็นฟังก์ชันการขึ้นต่อกัน เนื่องจากแถวข้อมูลมีค่าแอททริบิวต์ Weight และ Size เท่ากันแล้วสามารถบ่งชี้ค่าโดเมนของแอททริบิวต์ Mileage ค่าเดียวเสมอ

จากตัวอย่างที่ 3.8 แสดงให้เห็นว่า $R' \xrightarrow[F]{} D$ เป็นฟังก์ชันการขึ้นต่อกันแบบเต็ม เนื่องจากเซตย่อยทั้งหมด B คือ {Weight} และ {Size} ของเซตแอททริบิวต์ R' = {Weight, Size} ซึ่งเซตย่อยทั้งหมดเหล่านั้นไม่สามารถบ่งชี้แอททริบิวต์ D = {Mileage} ได้ ดังนั้น สรุปได้ว่า R' = {Weight, Size} เป็นแอททริบิวต์รีดัก

จากที่กล่าวมาในบทที่ 3 จะสรุปได้ว่านิยามแอททริบิวต์รีดักบนรัฟเซตเทียบเท่ากับกับนิยามแอททริบิวต์รีดักบนฟังก์ชันการขึ้นต่อกัน ซึ่งการค้นหาแอททริบิวต์รีดักบนรัฟเซตเทียบเท่ากับการค้นหาดีเทอร์มิแนนท์แบบเต็มซึ่งเป็นข้างซ้ายของฟังก์ชันการขึ้นต่อกัน โดยกำหนดให้แอททริบิวต์ตัดสินใจเป็นออบเจคดีเทอร์มิแนนท์แบบเต็มซึ่งเป็นข้างขวาของฟังก์ชันการขึ้นต่อกัน ดังนั้นจะกล่าวได้ว่าการค้นหาแอททริบิวต์รีดักทำได้จากการตรวจสอบฟังก์ชันการขึ้นต่อกันระหว่างเซตแอททริบิวต์ย่อยทั้งหมดของแอททริบิวต์เงื่อนไขกับแอททริบิวต์ตัดสินใจว่าเซตแอททริบิวต์ย่อยเซตใดบ้างที่มีคุณสมบัติเป็นฟังก์ชันการขึ้นต่อกันแบบเต็ม

บทที่ 4
อัลกอริทึมค้นหาแอททริบิวต์รีดัก

ในบทที่ 4 นี้ได้นำเสนออัลกอริทึมสำหรับการค้นหาแอททริบิวต์รีดักบนทฤษฎีฟังก์ชันการขึ้นต่อกัน โดยแบ่งเนื้อหาออกเป็น 3 ส่วน ดังนี้ ส่วนแรกกล่าวถึงอัลกอริทึมค้นหาแอททริบิวต์รีดักบนทฤษฎีฟังก์ชันการขึ้นต่อกัน ส่วนที่ 2 จะกล่าวถึงตัวอย่างการทำงานของอัลกอริทึมค้นหาแอททริบิวต์รีดักบนทฤษฎีฟังก์ชันการขึ้นต่อกัน และส่วนสุดท้ายเป็นการวิเคราะห์การทำงานของอัลกอริทึมค้นหาแอททริบิวต์รีดักบนทฤษฎีฟังก์ชันการขึ้นต่อกัน

4.1 อัลกอริทึม

ปัญหาการคำนวณหาแอททริบิวต์รีดักเป็นปัญหาการค้นหาโดยใช้วิธีการสร้างแล้วทดสอบ (Generate and Test) โดยหลักการค้นหาจะใช้วิธีการสร้างคำตอบที่เป็นไปได้แล้วทำทดสอบจนครบทุกคำตอบที่เป็นไปได้ในโครงข่ายคำตอบ ซึ่งวิธีการดังกล่าวจัดว่าเป็นวิธีการที่ไม่มีประสิทธิภาพ

จากการสังเกตพบว่า เมื่อมีการสร้างคำตอบที่เป็นไปได้ที่ประกอบด้วยแอททริบิวต์จำนวนน้อย หากพบว่าคำตอบนั้นเป็นแอททริบิวต์รีดักได้ จะกล่าวได้ว่า ซุปเปอร์เซต (Super Set) ของคำตอบทั้งหมดจะไม่เป็นแอททริบิวต์รีดัก ดังนั้น จึงสามารถตัดซุปเปอร์เซตของคำตอบนั้นทั้งหมดทิ้งได้ นอกจากนี้เมื่อมีการสร้างคำตอบที่เป็นไปได้ที่ประกอบด้วยแอททริบิวต์จำนวนมากแล้วพบว่าคำตอบนั้นไม่เป็นแอททริบิวต์รีดัก เซตย่อย (Subset) ของคำตอบที่เป็นไปได้ทั้งหมดจะไม่เป็นคำตอบแอททริบิวต์รีดักด้วยตามคุณสมบัติการขยายของฟังก์ชันการขึ้นต่อกัน ดังนั้น หากทำการตรวจสอบโครงข่ายคำตอบสองทิศทางจะได้อัลกอริทึมค้นหาที่มีประสิทธิภาพ อัลกอริทึมตามแนวคิดนี้แสดงดังนี้

Reduct Algorithm : Bidirectional Search

Input : Decision Table T(C, D) is consistent or $C \longrightarrow D$

Output : REDUCT_SET

```
1    i = 1
2    n = number of conditional attributes
3    FD_SET = {}, NFD_SET = {}, REDUCT_SET = C
4    While ( i ≤ ⌊n / 2⌋ )
5    {
6        Generate Candidate Set Level i and Prune;
7        if Candidate Set = {}
8        {
9          go to step 14
10       }
11       else
12       {
13           TestFD, ObtainFDandNFD, and UpdateReductSet;
14           If i ≠ n – i
15           {
16               Generate Candidate Set Level n-i and Prune;
17                   if Candidate Set = {}
18                   {
19                       go to step 27
20                   }
21                   else
22                   {
23                       TestFD,ObtainFDandNFD and UpdateReductSet;
24                   }
25           }
26       }
27       i = i+1
28   };
```

อัลกอริทึมนี้มีส่วนประกอบ 5 ส่วน ดังนี้

1. การสร้างเซตแอททริบิวต์รีดักแข่งขันระดับ i (Generate Candidate Set Level i)
2. การตัด (Prune)
3. การทดสอบฟังก์ชันการขึ้นต่อกัน (TestFD)
4. การเก็บผลลัพธ์จากการทดสอบฟังก์ชันการขึ้นต่อกัน (ObtainFDandNFD)
5. การเก็บผลลัพธ์แอททริบิวต์รีดัก (UpdateReductSet)

รายละเอียดแต่ละส่วนจะได้กล่าวถึงต่อไป

4.1.1 การสร้างเซตแอททริบิวต์รีดักแข่งขันระดับ i

การสร้างเซตแอททริบิวต์รีดักแข่งขันระดับ i คือ การนำแอททริบิวต์เงื่อนไขแต่ละตัวมาประกอบรวมกันเป็นเซตแอททริบิวต์ 1 เซตแอททริบิวต์ ซึ่งเซตแอททริบิวต์นี้มีจำนวนแอททริบิวต์เท่ากับ i ตัว ซึ่งเรียกเซตแอททริบิวต์นั้นว่าเซตแอททริบิวต์รีดักแข่งขัน ยกตัวอย่างเช่น สมมติให้ A, B, C, D, E เป็นแอททริบิวต์เงื่อนไข ดังนั้นจะสร้างเซตแอททริบิวต์รีดักแข่งขันในแต่ละระดับได้ดังนี้

Candidate Set Level 0 : {}

Candidate Set Level 1 : {A, B, C, D, E}

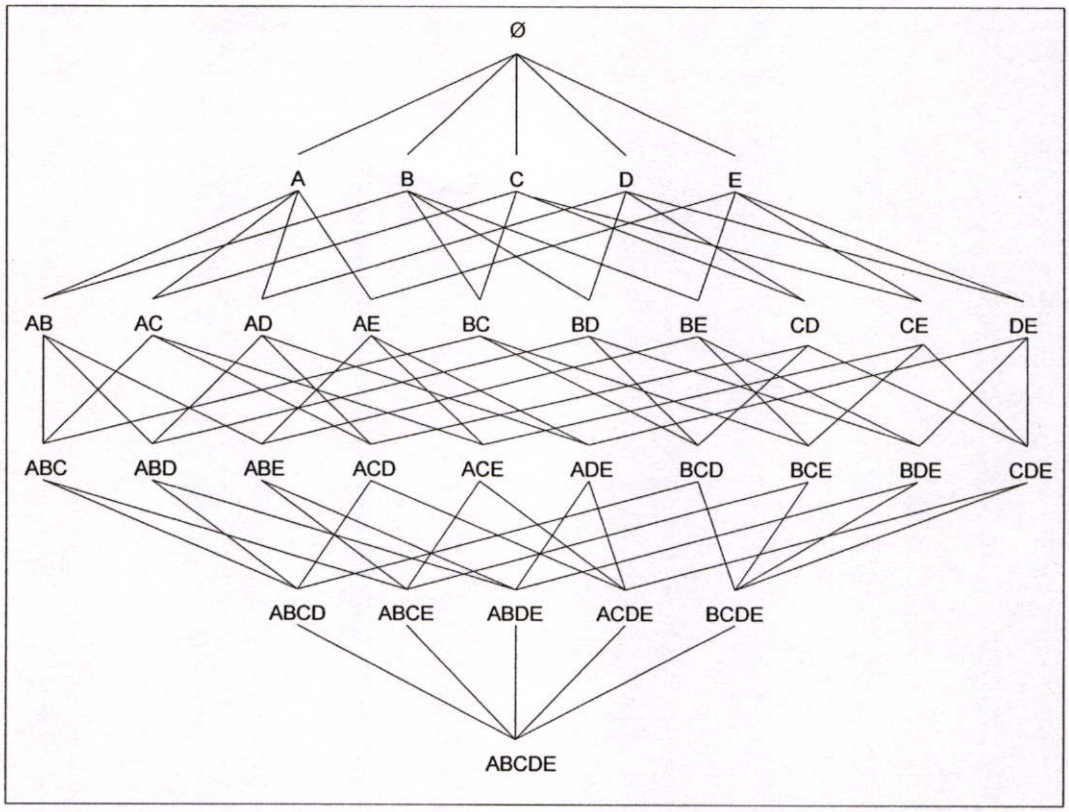
Candidate Set Level 2 : {AB, AC, AD, AE, BC, BD, BE, CD, CE, DE}

Candidate Set Level 3 : {ABC, ABD, ABE, ACD, ACE, ADE, BCD, BCE, BDE, CDE}

Candidate Set Level 4 : {ABCD, ABCE, ABDE, ACDE, BCDE}

Candidate Set Level 5 : {ABCDE}

แสดงดังรูปที่ 4.1



รูปที่ 4.1 แสดงเซตแอททริบิวต์รีดักแข่งขันระดับต่าง ๆ

4.1.2 การตัด

การตัดนั้นจะใช้คุณสมบัติฟังก์ชันการขึ้นต่อกันของกฎการขยายของอาร์มสตอง กล่าวคือ ถ้า $X \longrightarrow Y$ แล้ว $XZ \longrightarrow Y$ ซึ่งได้กล่าวคุณสมบัติดังกล่าวนั้นไว้ในบทที่ 3 และจากคุณสมบัตินี้จะพบว่า ถ้า $XZ \not\longrightarrow Y$ แล้ว $X \not\longrightarrow Y$ ด้วยเช่นกัน

ตัวอย่างที่ 4.1 กฎการขยายของอาร์มสตอง ถ้า $X \longrightarrow Y$ แล้ว $XZ \longrightarrow Y$

จากตารางที่ 4.1 แสดงให้เห็นว่า $A \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกัน เมื่อทำการพิจารณาซุปเปอร์เซตของแอททริบิวต์ A ซึ่งได้แก่ เซตแอททริบิวต์ AB, AC และ ABC ดังนี้

พิจารณาเซตแอททริบิวต์ $AB \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกันหรือไม่ จากตารางพบว่า $AB \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกัน

พิจารณาเซตแอททริบิวต์ $AC \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกันหรือไม่ จากตารางพบว่า $AC \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกัน

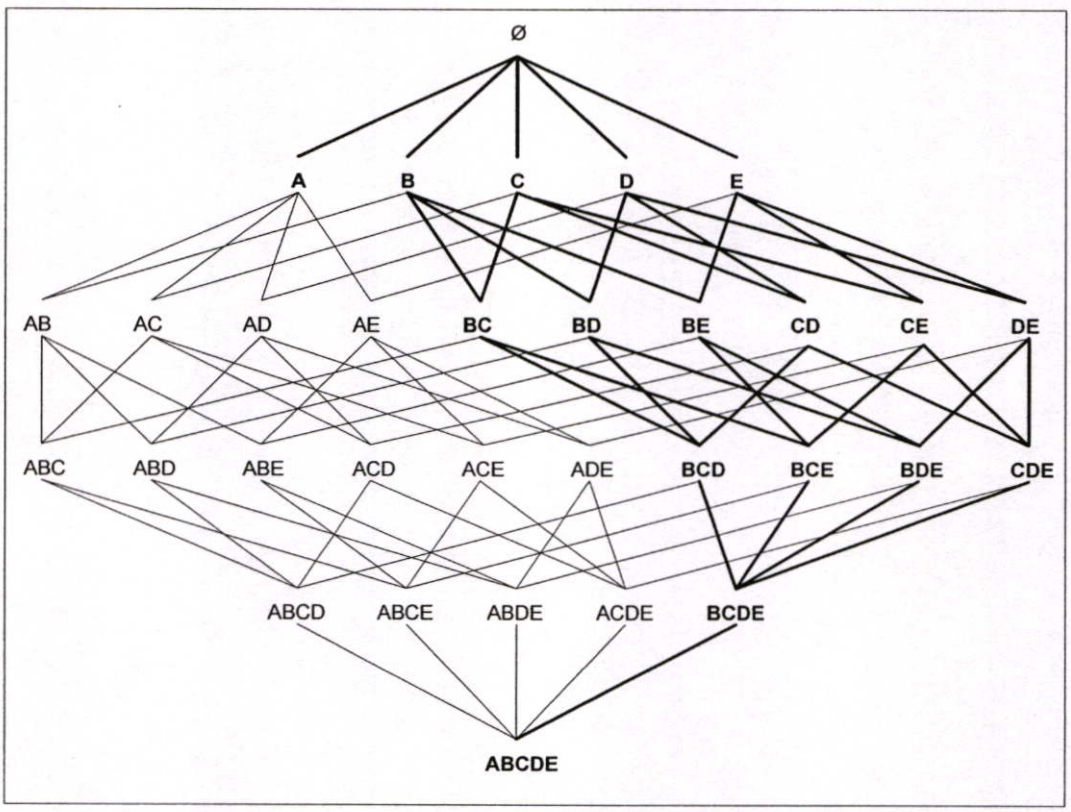
พิจารณาเซตแอททริบิวต์ $ABC \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกันหรือไม่ จากตารางพบว่า $ABC \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกัน

จากตัวอย่างแสดงให้เห็นว่าถ้าเซตแอททริบิวต์ใดบ่งชี้แอททริบิวต์อื่นเป็นฟังก์ชันการขึ้นต่อกันแล้วซุปเปอร์เซตทั้งหมดของเซตแอททริบิวต์เหล่านั้นจะบ่งชี้แอททริบิวต์อื่นเป็นฟังก์ชันการขึ้นต่อกันด้วย แต่เนื่องจากนิยามแอททริบิวต์รีดักได้กล่าวว่าแอททริบิวต์รีดักที่เหมาะสมที่สุดจะต้องเป็นแอททริบิวต์รีดักน้อยที่สุดที่สามารถบ่งชี้แอททริบิวต์ตัดสินใจได้ หรือกล่าวอีกนัยหนึ่งว่าเซตย่อยทั้งหมดของแอททริบิวต์รีดักไม่สามารถบ่งชี้แอททริบิวต์รีดักได้ ดังนั้นเราจะสามารถตัดซุปเปอร์เซตทั้งหมดของเซตแอททริบิวต์เหล่านั้นได้

ตารางที่ 4.1 ตารางฐานข้อมูลสัมพันธ์

Tuple	A	B	C	X
t_1	2	2	1	1
t_2	2	1	3	1
t_3	1	1	2	2
t_4	1	1	1	2
t_5	2	1	2	1
t_6	2	2	1	1

ตัวอย่างเช่นเมื่อทำการทดสอบว่าเซตแอททริบิวต์รีดักแข่งขันที่สร้างขึ้นเป็นฟังก์ชันการขึ้นต่อกัน แสดงให้เห็นว่าซุปเปอร์เซตของแอททริบิวต์รีดักแข่งขันไม่เป็นเซตแอททริบิวต์รีดัก สมมติว่า ถ้าในระดับ 1 พบว่า $A \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกัน ดังนั้น ในระดับ 2 ก็จะไม่ทำการสร้างเซตแอททริบิวต์ AB, AC, AD, AE ในระดับ 3 ก็จะไม่ทำการสร้างเซตแอททริบิวต์ ABC, ABD, ABE, ACD, ACE, ADE และเช่นเดียวกันในระดับ4 จะไม่ทำการสร้างเซตแอททริบิวต์ ABCD, ABCE, ABDE, ACDE เนื่องจากเซตแอททริบิวต์เหล่านี้เป็นซุปเปอร์เซตของแอททริบิวต์ A แสดงดังรูปที่ 4.2



รูปที่ 4.2 แสดงการตัดเส้นทางคำตอบแอททริบิวต์รีดักแข่งขัน

<u>ตัวอย่างที่ 4.2</u> กฎการขยายของอาร์มสตอง ถ้า $XZ \not\longrightarrow Y$ แล้ว $X \not\longrightarrow Y$

จากตารางที่ 4.2 แสดงให้เห็นว่าแถวที่ t_2 และแถวที่ t_5 มีค่าของแอททริบิวต์ A, B, C และ D เท่ากัน แต่แถวที่ t_2 และแถวที่ t_5 มีค่าแอททริบิวต์ X ไม่เท่ากัน ดังนั้น ABC ไม่เป็นฟังก์ชันการขึ้นต่อกัน เมื่อทำการพิจารณาเซตย่อยของแอททริบิวต์ ABC ซึ่งได้แก่ เซตแอททริบิวต์ A, B, C, AB, AC, BC ดังนี้

พิจารณาเซตแอททริบิวต์ $A \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกันหรือไม่ จากตารางพบว่าแถวที่ t_2 และแถวที่ t_5 มีค่าของแอททริบิวต์ A เท่ากับ 2 เท่ากัน แต่แถวที่ t_2 มีค่าแอททริบิวต์ X เท่ากับ 1 และแถวที่ t_5 มีค่าแอททริบิวต์ X เท่ากับ 5 ซึ่งมีค่าไม่เท่ากัน ดังนั้น A ไม่เป็นฟังก์ชันการขึ้นต่อกัน

พิจารณาเซตแอททริบิวต์ $B \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกันหรือไม่ จากตารางพบว่าแถวที่ t_2 และแถวที่ t_5 มีค่าของแอททริบิวต์ B เท่ากับ 1 เท่ากัน แต่แถวที่ t_2 มีค่าแอททริบิวต์ X เท่ากับ 1 และแถวที่ t_5 มีค่าแอททริบิวต์ X เท่ากับ 2 ซึ่งมีค่าไม่เท่ากัน ดังนั้น B ไม่เป็นฟังก์ชันการขึ้นต่อกัน

พิจารณาเซตแอททริบิวต์ $C \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกันหรือไม่ จากตารางพบว่าแถวที่ t_2 และแถวที่ t_5 มีค่าของแอททริบิวต์ C เท่ากับ 3 เท่ากัน แต่แถวที่ t_2 มีค่าแอททริบิวต์ X เท่ากับ 1 และแถวที่ t_5 มีค่าแอททริบิวต์ X เท่ากับ 2 ซึ่งมีค่าไม่เท่ากัน ดังนั้น C ไม่เป็นฟังก์ชันการขึ้นต่อกัน

พิจารณาเซตแอททริบิวต์ $AB \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกันหรือไม่ จากตารางพบว่าแถวที่ t_2 และแถวที่ t_5 มีค่าของแอททริบิวต์ A และ B เท่ากับ 2 และ 1 ตามลำดับซึ่งเท่ากัน แต่แถวที่ t_2 มีค่าแอททริบิวต์ X เท่ากับ 1 และแถวที่ t_5 มีค่าแอททริบิวต์ X เท่ากับ 2 ซึ่งมีค่าไม่เท่ากัน ดังนั้น AB ไม่เป็นฟังก์ชันการขึ้นต่อกัน

พิจารณาเซตแอททริบิวต์ $AC \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกันหรือไม่ จากตารางพบว่าแถวที่ t_2 และแถวที่ t_5 มีค่าของแอททริบิวต์ A และ C เท่ากับ 2 และ 3 ตามลำดับซึ่งเท่ากัน แต่แถวที่ t_2 มีค่าแอททริบิวต์ X เท่ากับ 1 และแถวที่ t_5 มีค่าแอททริบิวต์ X เท่ากับ 2 ซึ่งมีค่าไม่เท่ากัน ดังนั้น AC ไม่เป็นฟังก์ชันการขึ้นต่อกัน

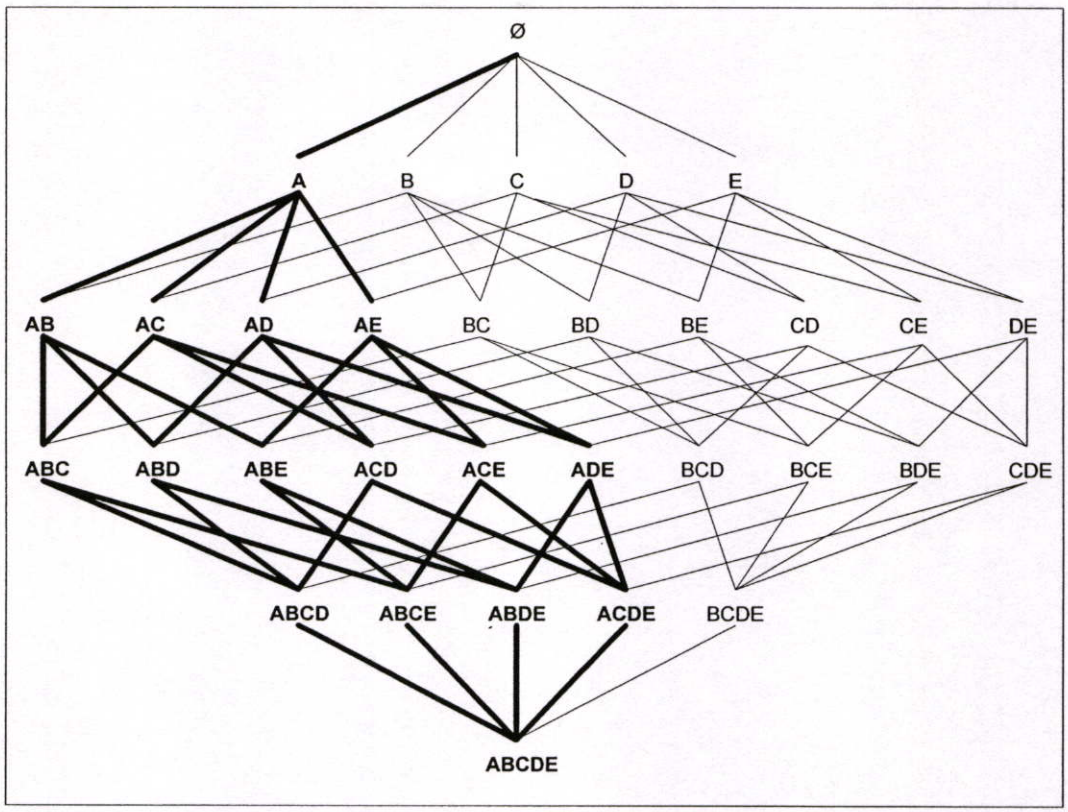
พิจารณาเซตแอททริบิวต์ $BC \longrightarrow X$ เป็นฟังก์ชันการขึ้นต่อกันหรือไม่ จากตารางพบว่าแถวที่ t_2 และแถวที่ t_5 มีค่าของแอททริบิวต์ B และ C เท่ากับ 1 และ 3 ตามลำดับซึ่งเท่ากัน แต่แถวที่ t_2 มีค่าแอททริบิวต์ X เท่ากับ 1 และแถวที่ t_5 มีค่าแอททริบิวต์ X เท่ากับ 2 ซึ่งมีค่าไม่เท่ากัน ดังนั้น BC ไม่เป็นฟังก์ชันการขึ้นต่อกัน

จากตัวอย่างแสดงให้เห็นว่าถ้าบางคู่แถวของเซตแอททริบิวต์ใดๆ ที่ทำให้เกิดเหตุการณ์วันทูเมนี่ ซึ่งจะทำให้ไม่เป็นฟังก์ชันการขึ้นต่อกันแล้วเซตย่อยทั้งหมดของเซตแอททริบิวต์นั้นจะทำให้เกิดเหตุการณ์วันทูเมนี่ที่คู่แถวนั้นด้วย หรือกล่าวได้ว่าเซตย่อยทั้งหมดเหล่านั้นไม่เป็นฟังก์ชันการขึ้นต่อกันด้วย ดังนั้น เราจะสามารถตัดเซตย่อยทั้งหมดของเซตแอททริบิวต์เหล่านั้นเนื่องจากแอททริบิวต์รีดักต้องมีคุณสมบัติฟังก์ชันการขึ้นต่อกันจึงจะสามารถเป็นแอททริบิวต์รีดักได้ซึ่งคุณสมบัติแอททริบิวต์รีดักนั้นได้กล่าวไว้ในบทที่ 3

ตารางที่ 4.2 ตารางฐานข้อมูลสัมพันธ์

Tuple	A	B	C	X
t_1	1	2	1	1
t_2	2	1	3	1
t_3	1	1	2	2
t_4	2	2	1	2
t_5	2	1	3	2
t_6	1	1	1	1

ตัวอย่างเช่นเมื่อทำการทดสอบว่าเซตแอททริบิวต์รีดักแข่งขันที่สร้างขึ้นไม่เป็นฟังก์ชันการขึ้นต่อกัน แสดงให้เห็นว่าแอททริบิวต์ย่อยของเซตแอททริบิวต์รีดักแข่งขันไม่เป็นเซตแอททริบิวต์รีดัก ถ้าในระดับ 4 ตรวจสอบพบว่า BCDE $\longrightarrow$ X ไม่เป็นฟังก์ชันการขึ้นต่อกัน ดังนั้น ในระดับที่ 1 ก็จะไม่ทำการสร้างเซตแอททริบิวต์ B, C, D, E ส่วนในระดับที่ 2 ก็จะไม่ทำการสร้างเซตแอททริบิวต์ BC, BD, BE, CD, CE, DE ในระดับ 3 จะไม่ทำการสร้างเซตแอททริบิวต์ BCD, BCE, BDE, CDE เนื่องจากเซตแอททริบิวต์เหล่านั้นเป็นเซตย่อยของ BCDE หรือกล่าวอีกนัยหนึ่งว่าถ้าเซตแอททริบิวต์รีดักแข่งขันที่สร้างขึ้นไม่เป็นฟังก์ชันการขึ้นต่อกัน แสดงให้เห็นว่าเซตย่อยของแอททริบิวต์รีดักแข่งขันไม่ใช่แอททริบิวต์รีดักที่น้อยที่สุด กล่าวคือ ไม่ใช่แอททริบิวต์รีดักที่เหมาะสมที่สุด ดังนั้น เซตย่อยของแอททริบิวต์ดังกล่าวไม่ควรถูกสร้างขึ้น แสดงดังรูปที่ 4.3



รูปที่ 4.3 แสดงการตัดเส้นทางคำตอบแอททริบิวต์รีดักแข่งขัน

สรุป ดังนั้นเราจะสามารถนำคุณสมบัติฟังก์ชันการขึ้นต่อกันดังกล่าวข้างต้นมาช่วยลดเส้นทางในการค้นหาคำตอบแอททริบิวต์รีดักน้อยที่สุด โดยจะทำการตัดเส้นทางที่ไม่สามารถเป็นคำตอบแอททริบิวต์รีดัก กล่าวคือ เราควรจะไม่ทำการสร้างเซตแอททริบิวต์ที่ไม่มีคุณสมบัติที่จะเป็นคำตอบแอททริบิวต์รีดักน้อยที่สุด หรือเราควรจะทำการตัดเส้นทางในการค้นหาคำตอบแอททริบิวต์รีดักเพื่อให้ได้รับคำตอบแอททริบิวต์รีดักน้อยที่สุดและค้นหาคำตอบได้รวดเร็วขึ้น รายละเอียดแนวคิดนี้แสดงในโปรซีเดอร์ Prune ดังต่อไปนี้

Procedure Prune (Candidate Set)

```
If NFD_SET ≠ {} or FD_SET ≠ {}
{
    For each element of Candidate Set
    {
        If it is subset of some elements of NFD
        {
            Delete it from Candidate Set
        }
        If it is superset of some elements of FD
        {
            Delete it from Candidate Set
        }
    }
};
```

4.1.3 การทดสอบฟังก์ชันการขึ้นต่อกัน

การทดสอบว่าเซตแอททริบิวต์รีดักแข่งขันบ่งชี้แอททริบิวต์ตัดสินใจหรือไม่ หรือทดสอบว่าเป็นฟังก์ชันการขึ้นต่อกันระหว่างหรือไม่ โดยสามารถใช้ภาษา SQL ดังกล่าวข้างต้นช่วยในการคำนวณ รายละเอียดของอัลกอริทึมแสดงในโปรซีเดอร์ Test_FD ดังนี้

Procedure Boolean TestFD (Candidate Set, Decision Attribute)

```
SELECT COUNT(*) INTO : ROW_COUNT
FROM T U, T V
WHERE U.Candidate = V.Candidate
AND U.Decision ≠ V.Decision;

If ROW_COUNT > 0
{
    return false
}
else
{
    return true
};
```

4.1.4 การเก็บผลลัพธ์จากการทดสอบฟังก์ชันการขึ้นต่อกัน

เซตแอททริบิวต์รีดักแข่งขันที่ถูกการทดสอบฟังก์ชันการขึ้นต่อกันแล้วว่าเซตแอททริบิวต์รีดักแข่งขันบ่งชี้แอททริบิวต์ตัดสินใจหรือไม่ ถ้าเซตแอททริบิวต์รีดักแข่งขันบ่งชี้แอททริบิวต์ตัดสินใจจะเก็บเซตแอททริบิวต์ไว้ใน FD_SET ถ้าเซตแอททริบิวต์รีดักแข่งขันไม่สามารถบ่งชี้แอททริบิวต์ตัดสินใจจะเก็บเซตแอททริบิวต์ไว้ใน NFD_SET เพื่อนำเซตแอททริบิวต์รีดักแข่งขันที่ถูกเก็บไว้ใน FD_SET หรือ NFD_SET มาช่วยในการตัดเส้นทางในการสร้างเซตแอททริบิวต์รีดักแข่งขันในเลเวลต่าง ๆ รายละเอียดของอัลกอริทึมนี้แสดงในโปรซีเดอร์ ObtainFDandNFD ดังนี้

Procedure ObtainFDandNFD(Candidate Set)

```
If Candidate Set is FD
{
    Add it to FD_SET
}
Else
{
    Add it to NFD_SET
};
```

4.1.5 การเก็บผลลัพธ์แอททริบิวต์รีดัก

การเก็บผลลัพธ์ทำโดยการอัพเดทคำตอบเซตแอททริบิวต์รีดัก โดยจะทำการพิจารณาจากการนำเซตแอททริบิวต์รีดักแข่งขันใน REDUCT_SET มาเปรียบเทียบกับเซตแอททริบิวต์รีดักแข่งขันใน FD_SET ถ้าเซตแอททริบิวต์รีดักแข่งขันใน REDUCT_SET ไม่มีสมาชิกบางตัวที่เหมือนกับเซตแอททริบิวต์รีดักแข่งขันใน FD_SET จะทำการเพิ่มเซตแอททริบิวต์รีดักแข่งขันใน FD_SET ลงใน REDUCT_SET ตัวอย่างเช่นเปรียบเทียบเซตแอททริบิวต์ AB กับ CD ปรากฏว่าสมาชิกของเซตแอททริบิวต์ AB ไม่มีแอททริบิวต์ C หรือ D ประกอบรวมอยู่ ดังนั้นจะเก็บเซตแอททริบิวต์ CD ใน REDUCT_SET แต่ถ้าเซตแอททริบิวต์รีดักแข่งขันใน REDUCT_SET มีสมาชิกบางตัวที่เหมือนกับเซตแอททริบิวต์รีดักแข่งขันใน FD_SET แล้วจะทำการตรวจสอบว่าเซตแอททริบิวต์รีดักแข่งขันใน REDUCT_SET เป็นซุปเปอร์เซตของเซตแอททริบิวต์รีดักแข่งขันใน FD_SET หรือไม่ ถ้าเป็นจะทำการลบเซตแอททริบิวต์รีดักแข่งขันใน REDUCT_SET ออกและทำการใส่เซตแอททริบิวต์รีดักแข่งขันใน FD_SET แทนที่ เพื่อจะทำให้คำตอบแอททริบิวต์รีดักที่ได้รับเป็นแอททริบิวต์รีดักที่น้อยที่สุด การอัพเดท REDUCT_SET จะทำทุกครั้งที่มีสมาชิกใหม่ของ FD_SET รายละเอียดแสดงในโปรซีเดอร์ Update_Reduct_Set ดังนี้

Procedure UpdateReductSet(Candidate Reduct in FD_SET)

If elements in REDUCT_SET $\cap$ element of Candidate Reduct in FD_SET $\neq \phi$

{

If there exists elements in REDUCT_SET that is superset of the Candidate Reduct

{

Delete the elements from REDUCT_SET

}

}

Add Candidate Reduct to REDUCT_SET;

4. 2 ตัวอย่าง

สมมติให้ ตารางตัดสินใจT(C, D) โดยที่ C = {A, B, C, D, E} และ D = {X}

ข้อมูลเข้า : ข้อมูลในตารางตัดสินใจเป็นข้อมูลคงที่ หรือ $C \longrightarrow D$

ขั้นตอนที่ 1 กำหนดให้ i=1

ขั้นตอนที่ 2 กำหนดให้ n=จำนวนแอททริบิวต์เงื่อนไขซึ่งในที่นี้ n=5

ขั้นตอนที่ 3 กำหนดให้ FD_SET = {}, NFD_SET = {}, REDUCT_SET = C

ขั้นตอนที่ 4 $1 \leq 2$ จริงทำการวนลูป While

ขั้นตอนที่ 6 สร้างเซตแอททริบิวต์รีดักแข่งขันในระดับ1 คือ {A, B, C, D, E} และนำเซตแอททริบิวต์รีดักแข่งขันที่สร้างขึ้นในขั้นตอนที่ 6 มาทำการตัด ปรากฏว่า FD_SET = {}, NFD_SET = {} ดังนั้นจะไม่ทำการตัดเซตแอททริบิวต์รีดักแข่งขัน

ขั้นตอนที่ 7 ตรวจสอบว่าเซตแอททริบิวต์รีดักแข่งขันเป็นเซตว่างหรือไม่ ปรากฏว่าไม่เป็นเซตว่าง ดังนั้น จะทำขั้นตอนที่ 13

ขั้นตอนที่ 13 นำเซตแอททริบิวต์รีดักแข่งขันทำการทดสอบฟังก์ชันการขึ้นต่อกัน ปรากฏว่ามี A เป็นฟังก์ชันการขึ้นต่อกันจริง ส่วน B, C, D, E ไม่เป็นฟังก์ชันการขึ้นต่อกัน เก็บผลลัพธ์จากการทดสอบ ดังนี้ FD_SET = {A} และ NFD_SET = {B, C, D, E} จากนั้นนำเซตแอททริบิวต์รีดักแข่งขันใน REDUCT_ SET มาเปรียบเทียบกับเซตแอททริบิวต์รีดักแข่งขันใน FD_SET ปรากฏว่าเซตแอททริบิวต์รีดักแข่งขันใน REDUCT_ SET เป็นซุปเปอร์เซตของเซตแอททริบิวต์รีดักแข่งขันใน FD_SET กล่าวคือ ABCD เป็นซุปเปอร์เซตของ A ดังนั้นจะทำการลบ ABCDE ออกจาก REDUCT_ SET และทำการนำ A ใส่ใน REDUCT_ SET แทน ดังนั้น REDUCT = {A}

ขั้นตอนที่ 14 ทำการตรวจสอบว่า $1 \neq 4$ จริง ดังนั้น ทำการสร้างเซตแอททริบิวต์ในขั้นตอนที่ 16 ต่อไปได้

ขั้นตอนที่ 16 สร้างเซตแอททริบิวต์รีดักแข่งขันในระดับ4 : {ABCD, ABCE, ABDE, ACDE, BCDE} นำเซตแอททริบิวต์รีดักแข่งขันที่สร้างขึ้นในขั้นตอนที่ 16 ทำการตัด ปรากฏว่า FD_SET = {A} ดังนั้น ABCD, ABCE, ABDE, ACDE จะถูกตัด

ขั้นตอนที่ 17 ตรวจสอบว่าเซตแอททริบิวต์รีดักแข่งขันไม่เป็นเซตว่าง ดังนั้นจะทำขั้นตอนที่ 23

ขั้นตอนที่ 23 นำเซตแอททริบิวต์รีดักแข่งขันที่เหลือจากการตัดในขั้นตอนที่ 16 มาทดสอบฟังก์ชันการขึ้นต่อกันต่อไปปรากฏว่า BCDE เป็นฟังก์ชันการขึ้นต่อกัน ดังนั้น FD_SET = {A, BCDE} จากนั้นทำการเปรียบเทียบเซตแอททริบิวต์ใน REDUCT_ SET มาเปรียบเทียบกับเซตแอททริบิวต์รีดักแข่งขันใน FD_SET ปรากฏว่าสมาชิกของ BCDE ไม่มีแอททริบิวต์ A ประกอบรวมอยู่ ดังนั้น REDUCT_SET จะถูกอัพเดท REDUCT_SET = {A, BCDE}

ขั้นตอนที่ 27 i=2 วนกลับไปขั้นตอนที่ 4

ขั้นตอนที่ 4 ตรวจสอบ $2 \leq 2$ จริงทำการวนลูป while

ขั้นตอนที่ 6 สร้างเซตแอททริบิวต์รีดักแข่งขันในระดับ 2 ได้ดังนี้ {AB, AC, AD, AE, BC, BD, BE, CD, CE, DE} นำเซตแอททริบิวต์รีดักแข่งขันที่สร้างขึ้นในขั้นตอนที่ 6 เข้าการตัด ปรากฏว่า FD_SET = {A, BCDE} ดังนั้น AB, AC, AD, AE จะถูกตัด

ขั้นตอนที่ 7 ตรวจสอบว่าเซตแอททริบิวต์รีดักแข่งขันไม่เป็นเซตว่าง เนื่องจากยังเหลือเซต {BC, BD, BE, CD, CE, DE} ดังนั้น จะทำขั้นตอนที่ 13

ขั้นตอนที่ 13 นำเซตแอททริบิวต์รีดักแข่งขันที่เหลือจากการตัด มาทดสอบฟังก์ชันการขึ้นต่อกันต่อไปปรากฏว่า BC เป็นฟังก์ชันการขึ้นต่อกัน และ BD, BE, CD, CE, DE ไม่เป็นฟังก์ชันการขึ้นต่อกัน ดังนั้น FD_SET = {A, BCDE, BC} และ NFD_SET = {B, C, D, BD, BE, CD, CE, DE} จากนั้นทำการเปรียบเทียบเซตแอททริบิวต์ใน REDUCT_ SET มาเปรียบเทียบกับเซตแอททริบิวต์รีดักแข่งขันใน FD_SET ปรากฏว่าสมาชิกของ BCDE มีแอททริบิวต์ BC ประกอบรวมอยู่และ BCDEเป็นซุปเปอร์เซตของ BC ดังนั้น REDUCT_SET = {A, BC}

ขั้นตอนที่ 14 ทำการตรวจสอบปรากฏว่า $2 \neq 3$ จริง

ขั้นตอนที่ 16 สร้างเซตแอททริบิวต์รีดักแข่งขันในระดับ3 ได้ดังนี้ {ABC, ABD, ABE, ACD, ACE, ADE, BCD, BCE, BDE, CDE} นำเซตแอททริบิวต์รีดักแข่งขันที่สร้างขึ้นเข้าสู้โปรซีเดอร์การตัดปรากฏว่า FD_SET = {A, BCDE, BC} ดังนั้น ABC, ABD, ABE, ACD, ACE ADE, BCD, BCE จะถูกตัด ส่วน NFD_SET = {B, C, D, BD, BE, CD, CE, DE} ไม่สามารถตัดได้ ดังนั้น จะเหลือเซตแอททริบิวต์รีดักแข่งขันเพียง 2 เซต ดังนี้ BDE, CDE

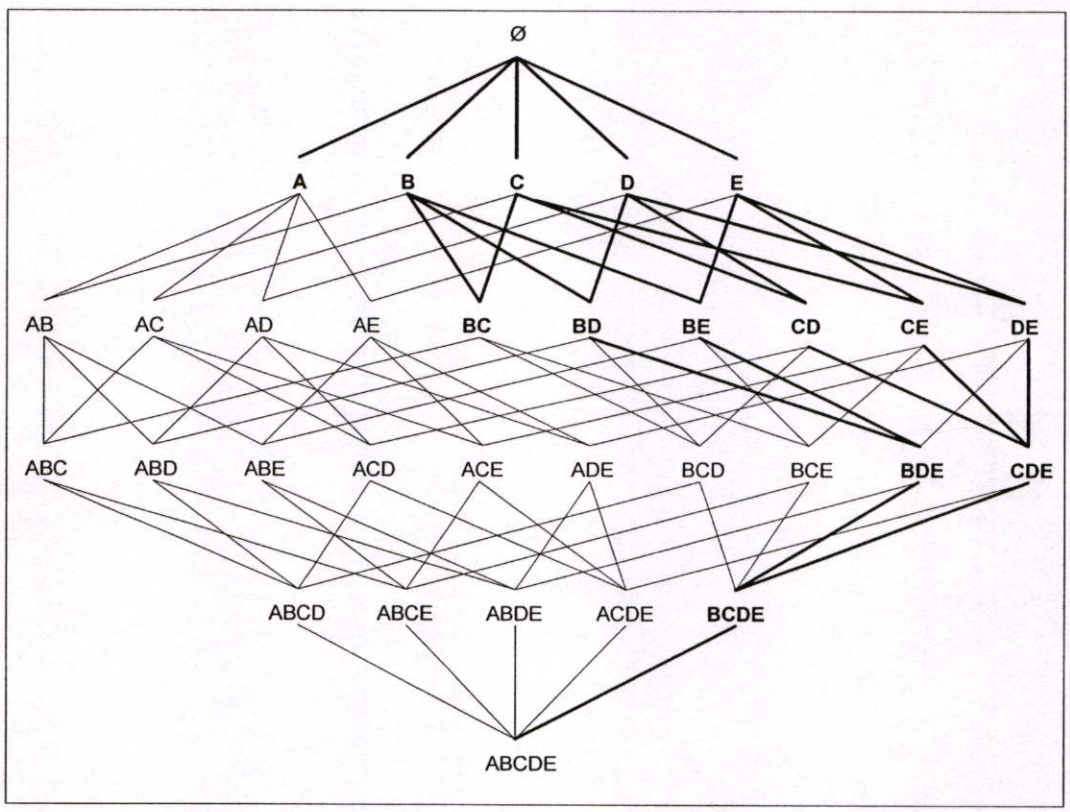
ขั้นตอนที่ 17 ตรวจสอบว่าเซตแอททริบิวต์รีดักแข่งขันไม่เป็นเซตว่าง ดังนั้น จะทำขั้นตอนที่ 23

ขั้นตอนที่ 23 นำเซตแอททริบิวต์รีดักแข่งขันที่เหลือจากการตัดในขั้นตอนที่ 16 มาทดสอบฟังก์ชันการขึ้นต่อกันต่อไปปรากฏว่า BDE, CDE ไม่เป็นฟังก์ชันการขึ้นต่อกัน ดังนั้น NFD_SET = {B, C, D, BD, BE, CD, CE, DE, BDE, CDE} จากนั้นทำการเปรียบเทียบเซตแอททริบิวต์ใน REDUCT_ SET มาเปรียบเทียบกับเซตแอททริบิวต์รีดักแข่งขันใน FD_SET ดังนั้น REDUCT_SET จะไม่ถูกอัพเดท REDUCT_SET = {A, BC}

ขั้นตอนที่ 27 i=3 วนกลับไปขั้นตอนที่ 4

ขั้นตอนที่ 4 ตรวจสอบ $3 \leq 2$ ไม่จริง ดังนั้น หยุดการวนลูป while

เมื่อวนครบทุกระดับจะสิ้นสุดการทำงานอัลกอริทึมนี้ ดังนั้น เซต REDUCT_SET = {A, BC} รายละเอียดการค้นหาในโครงข่ายคำตอบแสดงดังรูปที่ 4.4



รูปที่ 4.4 แสดงเส้นทางการหาคำตอบแอททริบิวต์รีดัก

4.3 การวิเคราะห์อัลกอริทึม

อัลกอริทึมจะมีการทำงาน 2 ทิศทาง คือ อัลกอริทึมจะเริ่มทำงานจากระดับต่ำสุดก่อน แล้วทำการสลับไปทำงานที่ระดับสูงสุด จากนั้นก็จะกลับมาทำงานระดับต่ำรองลงมา ก็จะกลับไปทำระดับสูงรองมา ซึ่งจะมีการทำงานแบบนี้ไปเรื่อย ๆ จนกว่าทำงานครบทุกระดับ เพื่อทำการสร้างเซตแอททริบิวต์รีดักแข่งขันในแต่ละระดับ จากนั้นอัลกอริทึมจะใช้คุณสมบัติของฟังก์ชันการขึ้นต่อกันในกฎการขยายของอาร์มสตองในการลดเส้นทางในการค้นหาแอททริบิวต์รีดัก โดยจะช่วยลดแอททริบิวต์รีดักแข่งขันที่ถูกนำมาสร้างในระดับต่าง ๆ จากนั้นทำการตรวจสอบว่าเซตแอททริบิวต์รีดักแข่งขันที่สร้างขึ้นถูกตัดออกหมดหรือไม่ ถ้าตัดออกหมดแสดงว่าเซตแอททริบิวต์รีดักแข่งขันเป็นเซตว่างจะทำให้อัลกอริทึมทำการสร้างเซตแอททริบิวต์รีดักแข่งขันระดับถัดไป แต่ถ้าเซตแอททริบิวต์รีดักแข่งขันไม่เป็นเซตว่างก็จะทำการตรวจสอบแต่ละเซตแอททริบิวต์รีดักแข่งขันที่เหลือในระดับนั้นว่ามีคุณสมบัติเป็นฟังก์ชันการขึ้นต่อกันหรือไม่ ถ้าเซตแอททริบิวต์รีดักแข่งขันเป็นฟังก์ชันการขึ้นต่อกันจะเก็บเซตแอททริบิวต์รีดักแข่งขันนั้นไว้ใน FD_SET แต่ถ้าเซตแอททริบิวต์รีดักแข่งขันไม่เป็นฟังก์ชันการขึ้นต่อกันจะเก็บเซตแอททริบิวต์รีดักแข่งขันนั้นไว้ใน NFD_SET เพื่อนำเซตแอททริบิวต์ที่เก็บไว้ใน FD_SET และ NFD_SET เป็นตัวช่วยตัดเส้นทางในการสร้างเซตแอททริบิวต์รีดักแข่งขันในระดับต่อไป จากนั้นถ้ามีการอัพเดทเซตแอททริบิวต์รีดักแข่งขันใน FD_SET ก็จะนำเซตแอททริบิวต์รีดักแข่งขันใน FD_SET นั้นมาเปรียบเทียบกับเซตแอททริบิวต์ใน REDUCT_SET เพื่อตรวจสอบว่ามีต้องอัพเดท REDUCT_SET หรือไม่ ถ้าเซตแอททริบิวต์รีดักแข่งขันใน REDUCT_SET ไม่มีสมาชิกบางตัวที่เหมือนกับเซตแอททริบิวต์รีดักแข่งขันใน FD_SET จะทำการเพิ่มเซตแอททริบิวต์รีดักแข่งขันใน FD_SET ลงใน REDUCT_SET แต่ถ้าเซตแอททริบิวต์รีดักแข่งขันใน REDUCT_SET มีสมาชิกบางตัวที่เหมือนกับเซตแอททริบิวต์รีดักแข่งขันใน FD_SET แล้วจะทำการตรวจสอบว่าเซตแอททริบิวต์รีดักแข่งขันใน REDUCT_SET เป็นซุปเปอร์เซตของเซตแอททริบิวต์รีดักแข่งขันใน FD_SET หรือไม่ ถ้าเป็นจะทำการลบเซตแอททริบิวต์รีดักแข่งขันใน REDUCT_SET ตัวนั้นออกและทำการใส่เซตแอททริบิวต์รีดักแข่งขันใน FD_SET แทนที่ เพื่อจะทำให้คำตอบแอททริบิวต์รีดักที่ได้รับเป็นแอททริบิวต์รีดักที่น้อยที่สุด ซึ่งอัลกอริทึมจะทำการวนสร้างเซตแอททริบิวต์ไปเรื่อย ๆ จนครบทุกระดับแล้วอัลกอริทึมจะทำการหยุดการทำงานของอัลกอริทึม

ความซับซ้อนของเวลา (Time Complexity) ที่ใช้ในการคำนวณของอัลกอริทึมนี้ขึ้นอยู่กับจำนวนสมาชิกของเซตแอททริบิวต์รีดักแข่งขันที่สร้างขึ้นทั้งหมด กรณีเลวร้ายที่สุดจะมีการสร้างสมาชิกของเซตแอททริบิวต์รีดักแข่งขันทั้งหมดเป็น $2^n - 1$ ตัว ดังนั้นความซับซ้อนของเวลาเป็น $O(2^n)$ กรณีดีที่สุด (Best Case) จะมีการสร้างสมาชิกของเซตแอททริบิวต์รีดักแข่งขันทั้งหมดเป็น n ตัว ดังนั้นความซับซ้อนของเวลาเป็น O(n)

สำหรับความซับซ้อนของเนื้อที่ (Space Complexity) ขึ้นอยู่กับขนาดของเซตแอททริบิวต์รีดักแข่งขันที่ใหญ่ที่สุด ซึ่งกำหนดโดย $C(n,m) = \frac{n!}{m!(n-m)!}$ โดยที่ $m = \lfloor med \rfloor$ และ med เป็นค่ามีเดียน (Median) ของ 1, 2, ..., n ดังนั้นความซับซ้อนของเนื้อที่สำหรับอัลกอริทึมนี้จึงเป็น O(C(n,m)) ด้วยความซับซ้อนนี้อัลกอริทึมรับประกันว่าคำตอบแอททริบิวต์รีดักไม่มีการสูญหาย หรือกล่าวได้ว่าจะทำให้ได้รับคำตอบแอททริบิวต์รีดักน้อยที่สุดครบทุกคำตอบ และได้รับคำตอบแอททริบิวต์รีดักถูกต้องเนื่องจากได้นำนิยามแอททริบิวต์รีดักใหม่มาใช้ในการสร้างอัลกอริทึม นอกจากนี้การตรวจสอบโครงข่ายสองทิศทางของการทำงานอัลกอริทึมจะทำให้ได้อัลกอริทึมค้นหาที่มีประสิทธิภาพ จะช่วยทำให้ลดเส้นทางของคำตอบแอททริบิวต์รีดักได้ดี ซึ่งคาดว่าจะทำให้ได้รับคำตอบแอททริบิวต์รีดักได้รวดเร็วขึ้น อย่างไรก็ตามในการตัดเส้นทางนั้นขึ้นอยู่กับข้อมูลเข้าเป็นสำคัญ ดังนั้น ถ้าข้อมูลเข้าส่งผลให้สามารถตัดเส้นทางของคำตอบแอททริบิวต์รีดักได้มากก็จะทำให้ได้รับคำตอบแอททริบิวต์รีดักได้เร็ว ส่วนการทดสอบฟังก์ชันการขึ้นต่อกันนั้นสามารถใช้ภาษา SQL เป็นตัวช่วยในการคำนวณ ซึ่ง SQL เป็นตัวปฏิบัติการฐานข้อมูลมีให้ใช้อยู่ในระบบฐานข้อมูลทั่วไปและในปัจจุบันภาษา SQL นั้นเป็นภาษาที่นิยมใช้ในการค้นหาข้อมูล (Query Search) ดังนั้น การคำนวณเพื่อทำการทดสอบฟังก์ชันการขึ้นต่อกันนั้นจะทำได้สะดวกและสามารถรองรับข้อมูลที่มีขนาดใหญ่ได้

บทที่ 5
สรุปและข้อเสนอแนะ

5.1 สรุป

งานวิจัยนี้เป็นการเชื่อมโยงระหว่างนิยามแอททริบิวต์รีดักบนรัฟเซตกับนิยามแอททริบิวต์รีดักบนฟังก์ชันการขึ้นต่อกัน โดยงานวิจัยได้นำเสนอนิยามใหม่ของแอททริบิวต์รีดักโดยอาศัยทฤษฎีฟังก์ชันการขึ้นต่อกันที่เทียบเท่ากับนิยามแอททริบิวต์รีดักบนรัฟเซต ซึ่งวิธีการสร้างนิยามใหม่นี้เริ่มจากการศึกษางานวิจัยที่เกี่ยวข้องกับนิยามแอททริบิวต์รีดักและนิยามต่างๆ บนทฤษฎีรัฟเซตและวิธีการค้นหาแอททริบิวต์รีดักบนรัฟเซต จากนั้นทำการศึกษาและวิเคราะห์นิยามแอททริบิวต์รีดักบนทฤษฎีรัฟเซต ซึ่งจากการวิเคราะห์พบว่านิยามแอททริบิวต์รีดักนี้เทียบเท่ากับทฤษฎีฟังก์ชันการขึ้นต่อกัน ดังนั้น จึงนำทฤษฎีฟังก์ชันการขึ้นต่อกันมาใช้ในการสร้างนิยามแอททริบิวต์รีดักใหม่ และเมื่อทำการสร้างนิยามใหม่ที่เทียบเท่ากับนิยามแอททริบิวต์รีดักบนรัฟเซตแล้วงานวิจัยได้นำนิยามแอททริบิวต์รีดักใหม่นี้สร้างอัลกอริทึมค้นหาแอททริบิวต์รีดักที่มีประสิทธิภาพ โดยได้นำคุณสมบัติของทฤษฎีฟังก์ชันการขึ้นต่อกันมาช่วยในการค้นหาแอททริบิวต์รีดัก ซึ่งอัลกอริทึมที่ได้ให้ผลลัพธ์ครบถ้วนและถูกต้อง ดังนั้น ผลลัพธ์ที่ได้รับจากการงานวิจัย คือ 1) นิยามใหม่ของแอททริบิวต์รีดักโดยอาศัยทฤษฎีฟังก์ชันการขึ้นต่อกัน และ 2) อัลกอริทึมใหม่แบบมีประสิทธิภาพสำหรับค้นหาแอททริบิวต์รีดักบนทฤษฎีฟังก์ชันการขึ้นต่อกัน

5.2 ข้อเสนอแนะ

แม้ว่างานวิจัยนี้ได้นำเสนอนิยามใหม่และอัลกอริทึมใหม่แบบมีประสิทธิภาพตามนิยามใหม่นี้ แต่ความมีประสิทธิภาพนี้ขึ้นอยู่กับลักษณะข้อมูลในตารางตัดสินใจเป็นสำคัญ ดังนั้น การทดลองความมีประสิทธิภาพควรทำบนข้อมูลจริงที่มีขนาดใหญ่เพื่อเทียบกับอัลกอริทึมอื่น ๆ โดยใช้ตัววัดประสิทธิภาพ เช่น จำนวนสมาชิกของเซตแอททริบิวต์แข่งขันที่มีการทดลอง เป็นต้น นอกจากการทดสอบประสิทธิภาพแล้ว งานวิจัยในอนาคตที่น่าสนใจ คือ การนำกฎการขยายของอาร์มสตองไปช่วยในการตัดเส้นทางการค้นหาในการสืบค้นหาฟังก์ชันการขึ้นต่อกันในฐานข้อมูลใด ๆ

เอกสารอ้างอิง

[1] Hong Y., Howard J.H., Cory J.B. **"FD_MINE: Discovering Functional Dependences in a Database Using Equivalence."** University of Regina, Computer Science Department, Technical Report CS-02-04, August 2002. ISBN 0-7731-0441-0

[2] Yka H., Juhu K., Pasi P., Hannu T. "TANE: An Efficient Algorithm for Discovering Functional and Approximate Dependencies." **The Computing Journal.,** vol. 42, No.2, 1999. pp.100-111.

[3] Huang Y., Shian-Shyong T., Wu G., Zhang F. "A Two-phase Feature Selection Method using both Filter and Wrapper." **IEEE SMC '99.,** vol.2, 12-15 Oct, 1999. pp.132 – 136.

[4] Jianchao H., Xiaohua H., Tsao Y.L. "Feature Subset Selection Based on Relative Dependency Between Attribute." **Springer-Verlag Heidelberg. On Comput.,** vol 3066, June 2004. pp. 176-185.

[5] Jing Z., Jianmin W., Deyi L., Huacan H., Jiaguang S. "A New Heuristic Reduct Algorithm Base on Rough Sets Theory." **Springer-Verlag Berlin Heidelberg.,** 2003. pp. 247–253.

[6] Jun Z., Guo-Yin W., Zhong-F.W., Hong T., Hua L. "The study on Technologies for feature selection." **IEEE Machine Learning and Cybernetics 2002.,** vol.2, 4-5 Nov, 2002. pp.689 – 693.

[7] Keyun H., Lili D., Yuchang L., C Shi. "A Heuristic Optimal Reduct Algorithm." **Springer-Verlag Heidelberg.,** vol. 1983, Dec 13-15 2000. pp.139-144.

[8] Keyun H., Lili D., Yuchang L., C Shi. **"Sampling for Approximate Reduct in very Large Datasets."** Computer Science Dapartment, Tsinghua University, Beijing 100084, P.R. China. 2000.

[9] Philip M.L, Arthur B., Michael K. **DATABASES AND TRANSACTION PROCESSING : An Application-Oriented Approach.** Copyright Addison Wesley. 2002

[10] Ron K., George H.J. "Wrappers for Feature Subset Selection." **Artificial Intelligence97.,** vol.1-2, May 20, 1997. pp. 273-324.

[11] Xiaohua H., Tsao Y.L.., Jiancho H. "A New Rough Sets Model Based on Database Systems." **Springer-Verlag Heidelberg, On Comput.,** vol. 2639, August 2003. pp. 114-121.

[12] **"Database Illuminated : Normalization."** [Online]. Available : http://cs3.wnmu.edu/Math&CS/mcfarland/CMPS465%20Adv%20DB%20PDF/Ricardo_ch05Slides.pdf

[13] **"Data Modeling."** [Online]. Available : http://www.utexas.edu/its/windows/database/datamodeling/rm/rm7.html

[14] **"Functional Dependencies."** [Online]. Available : http://www.cs.jcu.edu.au/Subjects/cp1500/1998/Lecture_Notes/normalisation/2nf.html

[15] **"Normal forms and Normalization."** [Online]. Available : http://user.it.uu.se/~udbl/dbt-ht2004/le4-normalizationx.pdf

ประวัติผู้เขียน

ชื่อ – สกุล	นางสาวดวงใจ วรเศรษฐเมธี
วัน เดือน ปีเกิด	15 ธันวาคม 2522
ที่อยู่	154/4 ถนนอัษฎางค์ เขตพระนคร แขวงวังบูรพาภิรมย์
	จังหวัดกรุงเทพมหานคร 10200
ประวัติการศึกษา	2544 วิทยาศาสตรบัณฑิต สาขาวิชาสถิติประยุกต์
	คณะวิทยาศาสตร์และเทคโนโลยี
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประวัติการทำงาน

มีนาคม – เมษายน 2543	ฝึกงานที่กองพัฒนางานคอมพิวเตอร์
	ณ ศาลาว่าการกรุงเทพมหานคร2
	เขตดินแดง จังหวัดกรุงเทพมหานคร
ตุลาคม 2546	ผู้ช่วยวิทยากร อบรมภาษาซี โครงการคอมพิวเตอร์โอลิมปิก
	ณ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
มิถุนายน-กันยายน 2548	อาจารย์พิเศษ มหาวิทยาลัยกรุงเทพ ณ วิทยาเขตรังสิต