การสร้างไวยากรณ์ไม่พึ่งบริบทแบบเชื่อมตรงโดยใช้ลำดับของกฎแฝง

นายสุรพงษ์ ผลประกอบศิลป์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2550

AN ON-LINE CONTEXT-FREE GRAMMARS CONSTRUCTION USING SEQUENCES OF HIDDEN PRODUCTIONS

Mr.Surapong Phonprakobsin

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science
Department of Computer Engineering
Chulalongkorn University
Academic Year 2007

หัวข้อวิทยานิพนธ์	การสร้างไวยากรณ์ไม่พึ่งบริบทแบบเชื่อมตรงโดยใช้ลำดับของกฎแฝง
โดย	นายสุรพงษ์ ผลประกอบศิลป์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์)

..... กรรมการ
(อาจารย์ ดร.เศรษฐา ปานงาม)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.อานนท์ รุ่งสว่าง)

นายสุรพงษ์ ผลประกอบศิลป์ : การสร้างไวยากรณ์ไม่พึ่งบริบทแบบเชื่อมตรงโดยใช้ลำดับของกฎแฝง (AN ON-LINE CONTEXT-FREE GRAMMARS CONSTRUCTION USING SEQUENCES OF HIDDEN PRODUCTIONS) อ.ที่ปรึกษา : ผศ.ดร.อรรถสิทธิ์ สุรฤกษ์, 38 หน้า

ในการวิเคราะห์และแก้ปัญหาด้านการอนุมานภาษาด้วยไวยากรณ์ไม่พึ่งบริบท งานวิจัยส่วนใหญ่จะมุ่งเน้นหาอัลกอริทึมเพื่อเรียนรู้และพัฒนาภาษาไวยากรณ์ไม่พึ่งบริบทจากการพิจารณาโครงสร้างของข้อมูลตัวอย่าง ซึ่งปัญหาที่พบคือ อัลกอริทึมส่วนใหญ่มีความซับซ้อนเชิงเวลาระดับเลขชี้กำลัง มีงานวิจัยของวุฒิ สุนทรภัณฑ์ ที่สามารถอนุมานภาษาโดยใช้ความซับซ้อนเชิงเวลาระดับพหุนาม ซึ่งต้องอาศัยตัวอย่างลบในการลดทอนความกว้างของภาษาขณะเรียนรู้

ดังนั้นงานวิจัยนี้จึงเสนออัลกอริทึมสร้างไวยากรณ์ไม่พึ่งบริบทแบบใหม่สำหรับบางภาษาไม่พึ่งบริบทรวมทั้งทุกภาษาสม่ำเสมอที่ใช้ความซับซ้อนเชิงเวลาระดับพหุนาม สามารถเรียนรู้ภาษาได้ด้วยจากตัวอย่างบวก ไม่จำเป็นต้องใช้ตัวอย่างลบ หลักการทำงานของอัลกอริทึม จะเริ่มสร้างไวยากรณ์ไม่พึ่งบริบทที่กว้างครอบคลุมทุกตัวอย่าง แล้วเรียนรู้รูปแบบลำดับของกฎแฝงให้อยู่ในรูปภาษาสม่ำเสมอ จะเห็นว่าลดความซับซ้อนเชิงเวลาน้อยกว่าเมื่อเทียบกับงานวิจัยอื่น และงานวิจัยของวุฒิ สุนทรภัณฑ์ นอกจากนี้การทำงานของอัลกอริทึมเป็นแบบเชื่อมตรง คือสามารถเรียนรู้รูปแบบลำดับของกฎแฝงไปเรื่อยๆ เมื่อมีตัวอย่างใหม่เข้ามา

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต สุรพงษ์ ผลประกอบศิลป์
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษา Aphant Lunamla
ปีการศึกษา 2550

4770516821 : MAJOR COMPUTER SCIENCE
KEY WORD : GRAMMATICAL INFERENCE / REGULAR LANGUAGE / CONTEXT-FREE GRAMMAR / EARLEY PARSING

SURAPONG PHONPRAKOBSIN : AN ON-LINE CONTEXT-FREE GRAMMARS CONSTRUCTION USING SEQUENCES OF HIDDEN PRODUCTIONS. THESIS ADVISOR : ASST. PROF. ATHASIT SURARERKS, Ph.D., 38 pp.

In an analysis and solving of grammar induction with context-free grammar, many researches focused on the algorithms that learned from considering the structure of data. The problem is that most algorithms have exponential time complexity. Wutthi Soonthonpant's research can inference languages in polynomial time complexity, but needs negative examples to reduce the redundant of language during the learning process.

Thus, this proposed research introduces a new construction algorithm for some context-free languages including all regular languages that has polynomial time complexity. This algorithm can be learned from only positive examples, with no need for negative examples. The concept of the algorithm is to initialize a general context-free grammar that covers all data, and learns sequences of hidden productions used in regular language. The time complexity of this algorithm is less than the other researches and Wutthi Soonthonpant's research. In addition, The proposed algorithm also uses an on-line algorithm that can learn pattern of hidden productions in real-time, when receiving new input data.

Department : Computer Engineering Student's signature : สุรพงษ์ ผลประกอบศิลป์
Field of study : Computer Science Advisor's signature : Athasit Surarerks
Academic year : 2007

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความอนุเคราะห์ และความช่วยเหลืออย่างยิ่งจาก ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์ อาจารย์ที่ปรึกษา ซึ่งเป็นผู้ให้ข้อคิด แนวทางและคำปรึกษา ตลอดจนเป็นผู้ตรวจทานแก้ไข ทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี ขอขอบพระคุณท่าน ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์ เป็นอย่างสูงที่ให้ความเมตตาช่วยเหลือ รวมทั้งโอกาสและสิ่งที่ดีแก่ผู้วิจัยเสมอมา

ขอขอบพระคุณ ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา ประธานกรรมการสอบวิทยานิพนธ์ อาจารย์ ดร.เศรษฐา ปานงาม ผู้ช่วยศาสตราจารย์ ดร.อานนท์ รุ่งสว่าง กรรมการสอบวิทยานิพนธ์ ที่ได้กรุณาให้คำแนะนำในการแก้ไขวิทยานิพนธ์ให้มีคุณภาพยิ่งขึ้น และขอขอบพระคุณคณาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัยทุกท่านที่ประสิทธิ์ประสาทความรู้อันมีค่ายิ่งแก่ผู้วิจัย

ท้ายที่สุดนี้ขอขอบพระคุณ บิดา มารดา ที่เป็นกำลังใจสำคัญ และขอขอบคุณเพื่อนๆ พี่ๆ และน้องๆ ทุกคน ที่ผลักดันและให้ความช่วยเหลือในทุกๆ ด้านจนผู้วิจัยสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วง

สารบัญ

หน้า

บทที่ หน้า

สารบัญตาราง

สารบัญภาพ

ภาพที่ หน้า

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันงานทางด้านวิศกรรมสารสนเทศ (information engineering) ได้แก่ การประมวลผลภาษาธรรมชาติ (natural language processing) และการวิเคราะห์ยีน (gene analysis) งานเหล่านี้ล้วนต้องอาศัยการรู้จำรูปแบบ (pattern recognition) เป็นหลักสำคัญ โดยมีงานวิจัยเกี่ยวกับการรู้จำรูปแบบมีอยู่หลายด้าน ได้แก่ งานวิจัยด้านการเรียนรู้ตัวแบบจำลองโครงข่ายประสาทเทียม (artificial neural network) และงานวิจัยด้านการเรียนรู้ด้วยอัลกอริทึมเชิงพันธุกรรม (genetic algorithm) ซึ่งล้วนอาศัยการเรียนรู้จากชุดข้อมูลตัวอย่างเพื่อคำนวณหาฟังก์ชันทางคณิตศาสตร์ที่สามารถนำมาใช้แยกแยะข้อมูลได้ ส่วนงานวิจัยด้านการอนุมานไวยากรณ์ (grammatical inference) คือ การเรียนรู้จากข้อมูลตัวอย่างเพื่อหากฎเกณฑ์ที่แสดงวากยสัมพันธ์ (syntax) โดยพิจารณาปัญหาให้อยู่ในรูปของไวยากรณ์ ซึ่งนอกจากจะสามารถนำไปใช้แยกแยะข้อมูลได้แล้ว ยังทำการตรวจจับรูปแบบโครงสร้างของข้อมูลที่เกิดขึ้นได้

ในปี 1967 โกลด์ [2] ได้เสนอทฤษฎีที่ว่า ความสามารถในการเรียนรู้ด้วยการอนุมานจากตัวอย่างที่อยู่ในระดับภาษาเวียนเกิดแบบแจงนับ (recursively enumerable language) แล้ว สามารถระบุตัวแทนของภาษาได้โดยจำกัด หรือเรียกว่าหลักการจำแนกภาษาภายในจำกัด (language identification in the limit) ซึ่งต้องอาศัยตัวช่วยในการตอบความเป็นสมาชิกของตัวอย่างในภาษา แต่ไม่ได้ระบุเวลาที่ใช้ในการคำนวณไว้ ถัดมาในปี 1997 อิกูรา [13] ได้พิสูจน์ว่า การระบุภาษาในระดับไม่พึ่งบริบทนั้นไม่สามารถทำได้ในความซับซ้อนเชิงเวลาระดับพหุนาม ถัดมาในปี 2005 นากามุระและมัตซุโมโต [3] ได้นำเสนออัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบท โดยพิจารณาจากข้อมูลตัวอย่างบวกและข้อมูลตัวอย่างลบ แต่ยังคงใช้ความซับซ้อนเชิงเวลาระดับเลขชี้กำลัง ถัดมาในปี 2006 วุฒิ สุนทรภัณฑ์ [4] ได้ลดความซับซ้อนในการคำนวณให้น้อยลงเพื่อให้การเรียนรู้ภาษาในระดับไม่พึ่งบริบทสามารถทำได้ในความซับซ้อนเชิงเวลาระดับพหุนาม โดยเสนอการปรับปรุงและพัฒนาอัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบทที่ใช้วิธีการแทนสายอักขระและผสานด้วยตัวแปร ร่วมกับการพิจารณาสร้างกฎวนซ้ำจากข้อมูลตัวอย่างบวก ในอัลกอริทึมที่วุฒิเสนอจะใช้ข้อมูลตัวอย่างลบเพื่อลดทอนกฎเกินจำเป็น เพื่อไม่ให้ไวยากรณ์ไม่พึ่งบริบทที่สร้างขึ้นกว้างมากเกินไป แต่ปัญหาที่พบคือ ในแต่ละรอบอัลกอริทึมใช้ความซับซ้อนในการคำนวณสูงในการตัดทอนกฎจากข้อมูลตัวอย่างลบ รวมทั้งต้องใช้ข้อมูลตัวอย่างจำนวนมากในการเรียนรู้เพื่อให้ไวยากรณ์ลู่เข้า

งานวิจัยนี้จึงนำเสนออัลกอริทึมสร้างไวยากรณ์ไม่พึ่งบริบทแบบใหม่ที่ใช้ความซับซ้อนเชิงเวลาระดับพหุนาม ที่เริ่มต้นจากการสร้างไวยากรณ์ไม่พึ่งบริบทที่ครอบคลุมข้อมูลทุกตัวอย่างบวกขึ้นมา แล้วเรียนรู้รูปแบบสายลำดับของกฎแฝง (hidden rules) ที่เกิดขึ้นแทน

โดยสายลำดับของกฎแฝงอาศัยการดึงลำดับของการใช้กฎที่ได้จากการแปลงของสายอักขระ (derivation) อัลกอริทึมดังกล่าวสามารถเรียนรู้โดยใช้เพียงข้อมูลตัวอย่างบวกเท่านั้น ไม่จำเป็นต้องใช้ข้อมูลตัวอย่างลบ ซึ่งช่วยลดความซับซ้อนในการคำนวณให้น้อยลงไปมาก นอกจากนี้ การทำงานของอัลกอริทึมเป็นแบบเชื่อมตรง คือ สามารถเรียนรู้ได้เรื่อยๆ เมื่อมีตัวอย่างใหม่เข้ามา โดยยังไม่พบงานวิจัยที่ใช้การเรียนรู้ไวยากรณ์ไม่พึ่งบริบทโดยใช้ลำดับของกฎแฝงดังกล่าว

1.2 วัตถุประสงค์ของการวิจัย

นำเสนอไวยากรณ์ไม่พึ่งบริบทสำหรับแยกแยะข้อมูลในภาษาจากการเรียนรู้รูปแบบลำดับของกฎแฝง ที่มีการทำงานเป็นแบบเชื่อมตรง

1.3 ขอบเขตของการวิจัย

1.3.1 เซตข้อมูลตัวอย่างที่ใช้ในการเรียนรู้ต้องเป็นภาษาในระดับไม่พึ่งบริบทเท่านั้น
1.3.2 ข้อมูลตัวอย่างใช้เพียงข้อมูลตัวอย่างบวกเท่านั้น
1.3.3 ไวยากรณ์ที่ได้อาศัยการบรรยายสม่ำเสมอร่วมพิจารณา
1.3.4 ความถูกต้องของไวยากรณ์ขึ้นกับจำนวนข้อมูลตัวอย่างที่ใช้ในการเรียนรู้
1.3.5 อัลกอริทึมที่ได้ต้องมีความซับซ้อนเชิงเวลาที่น้อยกว่า เมื่อเทียบกับอัลกอริทึมของวุฒิ สุนทรภัณฑ์ [4] และอัลกอริทึมของนากามุระและมัตซุโมโต [3]

1.4 ขั้นตอนและวิธีดำเนินการวิจัย

1.4.1 ศึกษาทำความเข้าใจเกี่ยวกับภาษาและไวยากรณ์ไม่พึ่งบริบท
1.4.2 ศึกษางานวิจัยที่เกี่ยวข้องกับการอนุมานไวยากรณ์
1.4.3 ออกแบบอัลกอริทึมสำหรับสร้างไวยากรณ์ไม่พึ่งบริบทแบบเชื่อมตรงโดยใช้ลำดับของกฎแฝง
1.4.4 ทดสอบวิธีการที่ได้นำเสนอ
1.4.5 เปรียบเทียบผลที่ได้จากการทดสอบ
1.4.6 สรุปผลการวิจัย พร้อมข้อเสนอแนะ และจัดทำรายงานวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย

ได้สร้างไวยากรณ์ไม่พึ่งบริบทแบบเชื่อมตรงโดยใช้ลำดับของกฎแฝง ที่ใช้ความซับซ้อนเชิงเวลาระดับพหุนาม การทำงานอาศัยข้อมูลตัวอย่างบวกเพียงอย่างเดียว

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้ ผู้วิจัยจะขอกล่าวถึงความรู้พื้นฐานรวมทั้งทฤษฎีที่เกี่ยวข้อง ได้แก่ ภาษารูปนัย ภาษาสม่ำเสมอ เครื่องจักรแบบจำกัดสถานะ คุณสมบัติของภาษาสม่ำเสมอ ไวยากรณ์ไม่พึ่งบริบท การเรียนรู้ด้วยการอุปนัย และการอนุมานไวยากรณ์ไม่พึ่งบริบท รวมทั้งงานวิจัยที่เกี่ยวข้องจะถูกกล่าวถึงเป็นลำดับสุดท้าย

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ภาษารูปนัย (formal language)

ให้คำจำกัดความของชุดตัวอักษร สายอักขระ ภาษา และอื่นๆ ไว้ดังนี้

นิยามที่ 2.1 *ชุดตัวอักษร (alphabet) หมายถึง เซตจำกัดของสัญลักษณ์ ที่เป็นหน่วยย่อยสุดไม่สามารถแบ่งแยกได้ นิยมใช้สัญลักษณ์แทนชุดตัวอักษรด้วย Σ และเรียกสมาชิกในชุดตัวอักษรว่า อักขระ หรือตัวอักษร (character)*

นิยามที่ 2.2 *สายอักขระ (string) หมายถึง ลำดับของอักขระ ถ้าลำดับมีจำนวนอักขระเป็นจำนวนจำกัด สายอักขระนั้นจะถูกเรียกว่า สายอักขระจำกัด (finite string) แต่ถ้าลำดับมีจำนวนอักขระเป็นอนันต์ สายอักขระนั้นจะถูกเรียกว่า สายอักขระอนันต์ (infinite string)*

นิยามที่ 2.3 *ภาษา (language) หมายถึง เซตของสายอักขระที่มีจำนวนของอักขระเป็นจำนวนจำกัด โดยสมาชิกในภาษาจะถูกเรียกว่า คำ (word)*

นิยามที่ 2.4 *ภาษารูปนัย หมายถึง ภาษาที่คำต่างๆ ในภาษามีกฎหรือกติกาที่ชัดเจนในการพิจารณาว่าเป็นสมาชิกในภาษานั้นๆ โดยปราศจากความกำกวม นอกจากนี้ จำนวนของกฎหรือกติกาต้องมีเป็นจำนวนจำกัดด้วย*

นิยามที่ 2.5 *สายอักขระที่ไม่มีอักขระเลยจะถูกเรียกว่า สายอักขระว่าง (null string or empty string) นิยมใช้สัญลักษณ์แทนด้วย ε*

ในกรณีที่ภาษามีจำนวนคำเป็นอนันต์ ภาษานั้นจะถูกเรียกว่าเป็น ภาษาอนันต์ (infinite language) ไม่เช่นนั้นจะเรียกว่าเป็น ภาษาจำกัด (finite language)

นิยามที่ 2.6 *ความยาว (length) ของสายอักขระ หมายถึง จำนวนของอักขระที่ประกอบอยู่ในสายนั้น สำหรับสายอักขระว่าง เราจะถือว่า ความยาวจะมีค่าเป็น 0 เสมอ*

นิยามที่ 2.7 *ตัวดำเนินการคลีนสตาร์ (Kleene's star) นิยามบนเซต S ใดๆ หมายถึง*

$$S^* = \bigcup_{i=0}^{\infty} S^i$$

โดยที่ $S^i = \{x_1 x_2 x_3 ... x_i \mid \forall x_j \in S\}$

ตัวดำเนินการคลีนสตาร์ หรือบางทีเรียกว่า ตัวดำเนินการปิดของคลีน (Kleene's closure) เป็นตัวดำเนินการหนึ่งที่นิยมใช้กันในการกำหนดภาษา และจะเห็นว่า ภาษาที่เกิดจากตัวดำเนินการนี้จะสามารถเป็นเซตอนันต์ได้

นิยามที่ 2.8 *ตัวดำเนินการบวก (positive closure) นิยามบนเซต S ใดๆ หมายถึง*

$$S^+ = \bigcup_{i=1}^{\infty} S^i$$

โดยที่ $S^i = \{x_1 x_2 x_3 ... x_i \mid \forall x_j \in S\}$
จากนิยาม เราพอสังเกตได้ว่า สำหรับเซต S ใดๆ จะได้ว่า $S^+ \subseteq S^*$

ภาษาที่มีขนาดใหญ่ที่สุดคือ ภาษาที่ประกอบด้วยคำทุกคำที่สามารถสร้างจากชุดตัวอักษร Σ สามารถเขียนแทนด้วย Σ^* เช่น $\Sigma = \{0,1\}$

$$\Sigma^* = \{0,1\}^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, ...\}$$

ดังนั้นทุก L ที่เกิดขึ้นจาก Σ จะได้ว่า $L^+ \subseteq \Sigma^*$

นิยามที่ 2.9 *ฟังก์ชันย้อนกลับ (reverse function) ของสายอักขระใดๆ หมายถึง การเรียงสลับย้อนกลับของลำดับของอักขระที่ปรากฏในสายอักขระนั้น ในที่นี้จะใช้สัญลักษณ์ $reverse\,(x)$ แทนฟังก์ชันย้อนกลับของสายอักขระ x*

ภาษาพาลินโดรม (palindrome language) หมายถึง ภาษาที่มีสมาชิกเป็นคำที่เป็นไปได้ทั้งหมดจากชุดของตัวอักษรที่กำหนดมาให้และมีคุณสมบัติที่ว่า ฟังก์ชันย้อนกลับของคำนั้นมีค่าเท่ากับตัวมันเอง และเราจะเรียกว่าเป็นภาษาพาลินโดรมนิยามบนชุดตัวอักษรนั้น เราสามารถกำหนดได้ดังนี้

$$palindrome\ over\ \Sigma = \{x \in \Sigma^* \mid x = reverse\,(x)\}$$

ตัวอย่างเช่น

$$palindrome\ over\ \{a,b\} = \{\varepsilon, a, b, aa, bb, aaa, aba, bab, bbb, aaaa, ...\}$$

2.1.2 ภาษาสม่ำเสมอ (regular language)

เราสามารถนิยามตัวดำเนินการอธิบายภาษาสม่ำเสมอได้ดังนี้

นิยามที่ 2.10 *ตัวดำเนินการคลีนสตาร์บนตัวอักขระ a เขียนแทนด้วย a^* หมายถึง เซตของสายอักขระที่เกิดจากการเรียงต่อกันของ a จำนวนเท่าใดก็ได้*

$$a^* = \{\varepsilon, a, aa, aaa, aaaa, ...\}$$

นิยามที่ 2.11 *ตัวดำเนินการบวกของตัวอักขระ a และ b เขียนแทนด้วย $a+b$ หมายถึง เซตของสายอักขระที่เกิดจากการเลือกตัวอักขระ a หรือตัวอักขระ b อย่างใดอย่างหนึ่ง*

$$a+b = \{a, b\}$$

นิยามที่ 2.12 *การบรรยายสม่ำเสมอ (regular expression)*

กำหนดให้ Σ เป็นชุดตัวอักษร และการบรรยายสม่ำเสมอหมายถึงข้อกำหนดต่อไปนี้

1. *ทุกสมาชิกใน Σ เป็นการบรรยายสม่ำเสมอ*
2. *อักขระว่าง ε เป็นการบรรยายสม่ำเสมอ*
3. *สำหรับ x และ y ใด ๆที่เป็นการบรรยายสม่ำเสมอแล้วจะได้ว่า $(x), x+y, xy$ และ x^* เป็นการบรรยายสม่ำเสมอด้วย*
4. *ใช้เฉพาะกฎ 3 ข้อข้างต้นเท่านั้นในการสร้างการบรรยายสม่ำเสมอ*

นิยามที่ 2.13 *ภาษาใดที่สามารถเขียนบรรยายได้ด้วยการบรรยายสม่ำเสมอ จะเรียกภาษานั้นว่าเป็น ภาษาสม่ำเสมอ*

อนึ่งภาษาหนึ่ง ๆนั้น อาจสามารถเขียนการบรรยายสม่ำเสมอได้มากกว่าหนึ่งแบบ เนื่องจากการเขียนบรรยายแบบนี้ สามารถลดรูปหรือเปลี่ยนรูปได้ เช่น

$$(0^*1^*)^* = (0+1)^*$$

$$(0+1)^* 01(0+1)^* + 1^*0^* = (0+1)^*$$

$$1^*1^* = 1^*$$

$(\varepsilon + 1)1^* = 1^*$ เป็นต้น

ทฤษฎีบทที่ 2.1 *ภาษาจำกัดทุกภาษาสามารถเขียนบรรยายด้วยการบรรยายสม่ำเสมอได้เสมอ*

2.1.3 เครื่องจักรสถานะแบบจำกัด (finite state machine)

การตรวจสอบการเป็นสมาชิกในภาษา สามารถทำได้โดยใช้ตัวแบบทางคณิตศาสตร์ (mathematical model) ที่เป็นที่นิยม และเป็นตัวแบบอย่างง่าย (simple model) ที่เรียกว่า เครื่องจักรจำกัดสถานะ หรือบางทีเรียกว่า ออโตมาตา (automata) โดยสนใจวิธีในการสร้างตัวแบบที่เหมาะสมกับภาษาที่กำหนดมาให้ ดังนั้นตัวแบบนี้จะตอบคำถามได้เพียงสองรูปแบบเท่านั้น คือ เป็นสมาชิกหรือไม่เป็นสมาชิก

นิยามที่ 2.14 *เครื่องจักรออโตมาตาแบบจำกัด (Finite Automaton: FA) หรือ เครื่องจักรสถานะแบบจำกัด M ประกอบด้วยส่วนสำคัญ 5 ส่วน คือ*

$$M = (Q, \Sigma, q_0, A, \delta)$$

โดยที่ Q *เป็นเซตจำกัดของสถานะ (state) ของเครื่องจักร*

Σ *เป็นเซตจำกัดของข้อมูลนำเข้า ที่เรียกว่าชุดตัวอักษร*

q_0 *เป็นสถานะเริ่มต้น (initial state) และเป็นสมาชิกใน Q*

A *เป็นเซตของสถานะของการยอมรับ (accepted state) หรือสถานะจบ (final state) และ $A \subseteq Q$ เสมอ*

δ *เป็นฟังก์ชันการเปลี่ยนสถานะ หรือเรียกว่า ฟังก์ชันการผ่าน (transition function) ที่กำหนดโดย $\delta : Q \times \Sigma \to Q$*

การทำงานของออโตมาตานั้น จะทำงานโดยการรับข้อมูลนำเข้าที่เป็นสายอักขระโดยการอ่านข้อมูลทีละอักขระตามลำดับ ทั้งนี้เพราะออโตมาตาเป็นเครื่องจักรที่มีการทำงานแบบลำดับ (sequential machine) โดยในตอนเริ่มต้นก่อนการทำงาน ออโตมาตาจะอยู่ในสถานะเริ่มต้น เมื่อข้อมูลนำเข้าถูกอ่านทีละอักขระ ออโตมาตาจะเปลี่ยนสถานะของเครื่องจักรไปตามข้อมูลที่อ่านเข้ามานั้น ซึ่งการเปลี่ยนสถานะนี้ได้ถูกกำหนดไว้แล้วโดยฟังก์ชันการเปลี่ยนสถานะ จากนั้นเริ่มต้นจากสถานะที่เป็นอยู่ ออโตมาตาจะดำเนินการอ่านต่อไปทีละอักขระพร้อมกับการพิจารณาเปลี่ยนสถานะไปตามฟังก์ชัน จนกระทั่งอักขระสุดท้ายถูกอ่านไปเรียบร้อยแล้ว ออโตมาตาจะหยุดทำงาน สถานะสุดท้ายที่ออโตมาตาหยุดการทำงานจะมีผลต่อคำตอบ เพราะถ้าออโตมาตาหยุดที่สถานะที่เป็นสถานะของการยอมรับ เราจะเรียกว่า ออโตมาตานั้นยอมรับข้อมูลนำเข้า ไม่เช่นนั้นจะเรียกว่า ออโตมาตานั้นปฏิเสธข้อมูลนำเข้า ดังนิยามต่อไปนี้

นิยามที่ 2.15 *สายอักขระที่ทำให้ออโตมาตาหยุดที่สถานะยอมรับ จะถูกเรียกว่า สายอักขระนั้นถูกยอมรับโดยออโตมาตา (accepted by automaton) ไม่เช่นนั้นจะถูกเรียกว่าถูกปฏิเสธโดยออโตมาตา (rejected by automaton)*

2.1.4 คุณสมบัติภาษาสม่ำเสมอ

ในที่นี้จะกล่าวถึงคุณสมบัติบางประการของภาษาสม่ำเสมอ ภาษาที่เป็นภาษาสม่ำเสมอนั้นจะต้องมีออโตมาตาที่ยอมรับ ซึ่งออโตมาตามีจำนวนสถานะจำกัด แต่ภาษาสม่ำเสมอนั้นอาจมีจำนวนสถานะอนันต์ได้ ทำให้เห็นว่าสำหรับคำใดๆก็ตามที่อยู่ในภาษา ถ้าคำนั้นมีความยาวเกินกว่าหรือเท่ากับจำนวนสถานะในออโตมาตาแล้ว จะมีต้องมีการวงจร (circuit) เกิดขึ้นในเส้นทาง (path) ที่ยอมรับคำนั้น คุณสมบัติที่ว่านี้เองที่ทุกภาษาสม่ำเสมอต้องมี

รวมทั้งในอีกความหมายของภาษาสม่ำเสมอที่ได้กล่าวในนิยามที่ 2.13 ไว้ว่า ภาษาสม่ำเสมอ หมายถึง ภาษารูปนัยที่สามารถเขียนบรรยายได้ด้วยการบรรยายแบบสม่ำเสมอได้

ทฤษฎีบทที่ 2.2 *กำหนดให้* L_1 *และ* L_2 *เป็นภาษาสม่ำเสมอสองภาษา พิจารณาได้ว่า*

$$L_1 \cup L_2$$
$$L_1 L_2$$
$$(L_1)^*$$

ทั้งสามภาษาข้างต้นเป็นภาษาสม่ำเสมอ

ทฤษฎีบทที่ 2.3 *สำหรับภาษา* L *ที่เป็นภาษาสม่ำเสมอและยอมรับได้โดยออโตมาตาจำกัดสถานะเชิงกำหนด (Deterministic Finite state Automata : DFA) ที่มีจำนวนสถานะเท่ากับ* n *สำหรับสายอักขระ* x *ในภาษา* L *ที่* $|x| \geq n$ *แล้ว* x *สามารถที่จะเขียนได้เป็น* $x = uvw$ *สำหรับบาง* u, v *และ* w *ที่สอดคล้องกับ*

$$|uv| \geq n$$
$$|v| \geq 0$$

สำหรับทุกจำนวนเต็ม $m \geq 0$ *จะได้ว่า* $uv^m w$ *เป็นสมาชิกในภาษาด้วย*

2.1.5 ไวยากรณ์ไม่พึ่งบริบท (context-free grammar)

ไวยากรณ์ไม่พึ่งบริบท เป็นวิธีการบรรยายภาษาด้วยกฎไวยากรณ์ที่ใช้ในการสร้างเซตของสายอักขระที่มีจำนวนกฎจำกัด ด้วยวิธีการสร้างแบบวนซ้ำ (recursive method) ซึ่งเป็นกฎเพียงกฎเดียวที่สามารถจะทำการนำมาใช้ได้หลายๆ ครั้ง ซึ่งสามารถนิยามความหมายได้ดังต่อไปนี้

นิยามที่ 2.16 *ไวยากรณ์ไม่พึ่งบริบท หมายถึง การบรรยายภาษาด้วยกฎไวยากรณ์ ซึ่งประกอบด้วยส่วนสำคัญ 4 ส่วน ดังนี้*

$$G = (V, \Sigma, P, S)$$

โดยที่ V *เซตจำกัดของตัวแปร (finite set of variable or non-terminal)*

Σ *เซตจำกัดของตัวอักษร (finite set of alphabet)*

S *ตัวแปรเริ่มต้น (start variable)* $\in V$

P *เซตจำกัดของกฎไวยากรณ์ (finite set of grammar rules or productions) เป็นเซตจำกัดที่อยู่ในรูปของ* $A \to \alpha$ *เมื่อ* $A \in V$ *และ* $\alpha \in (V \cup \Sigma)^*$

นิยามที่ 2.17 *ภาษาไม่พึ่งบริบท หมายถึง ภาษาที่สัมพันธ์กับไวยากรณ์ไม่พึ่งบริบท* G *หมายถึงเซตของสายอักขระที่มีคุณสมบัติดังนี้*

$$L(G) = \{x \in \Sigma^* \mid S \Rightarrow_G^* x\}$$

ตัวอย่างสายอักขระของภาษาที่อยู่ในระดับภาษาไม่พึ่งบริบท เช่น $L = \{a^n b^n \mid n \in Z^+$ และ $a, b \in \Sigma\}$ สามารถสร้างไวยากรณ์ไม่พึ่งบริบท G ที่บรรยาย L ได้ดังนี้

$$G = (V, \Sigma, P, S)$$

$$V = \{S\}$$

$$\Sigma = \{a, b\}$$

$$P = \{S \to aSb, S \to ab\}$$

ตัวอย่างการแปลง (derivation) ให้ได้ $aaabbb$ ที่เป็นสมาชิกของ $G = (V, \Sigma, P, S)$

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb$$

นิยามที่ 2.18 *สายอักขระ* $x \in (V \cup \Sigma)^*$ *ใดๆ ซึ่งมีตัวแปรปนกับตัวอักษรนั้น เราจะเรียกว่า รูปประโยค (sentential form)*

นิยามที่ 2.19 *ไวยากรณ์ไม่พึ่งบริบทเชิงเส้น (linear context-free grammar) คือ ไวยากรณ์ไม่พึ่งบริบทที่* P *เป็นเซตจำกัดที่อยู่ในรูปของ* $A \to \alpha$ *เมื่อ* $A \in V$ *และ* $\alpha \in (\Sigma^* V \Sigma^* \cup \Sigma^*)$

นิยามที่ 2.20 *ไวยากรณ์ไม่พึ่งบริบทเชิงเส้นแบบคู่ (even linear context-free grammar) คือ ไวยากรณ์ไม่พึ่งบริบทที่* P *เป็นเซตจำกัดที่อยู่ในรูปของ* $A \to \alpha$ *เมื่อ* $A \in V$ *และ* $\alpha \in (\Sigma V \Sigma \cup \Sigma \cup \varepsilon)$

นิยามที่ 2.21 *ไวยากรณ์เชิงเส้นเชิงกำหนด (deterministic linear grammar) คือ ไวยากรณ์ไม่พึ่งบริบทที่ P เป็นเซตจำกัดที่อยู่ในรูปของ $A \to \alpha$ เมื่อ $A \in V$ และ $\alpha \in (\Sigma V \Sigma^* \cup \varepsilon)$*

นิยามที่ 2.22 *ไวยากรณ์ไม่พึ่งบริบท G ใดๆเป็นไวยากรณ์ที่ไม่มีความกำกวม (unambiguous grammar) ก็ต่อเมื่อทุกสายอักขระที่เป็นสมาชิกของภาษา $L(G)$ สามารถทำการแปลงตัวแปรจากทางซ้ายสุดก่อนได้เพียงแบบเดียว แต่ถ้ามีคำที่เป็นสมาชิกของภาษา $L(G)$ สามารถทำการแปลงตัวแปรจากทางซ้ายสุดได้มากกว่าหนึ่งแบบ โดยใช้ลำดับการแปลงที่แตกต่างกัน จะเรียกไวยากรณ์นี้ว่า ไวยากรณ์ที่มีความกำกวม (ambiguous grammar)*

ตัวอย่างเช่น จากกฎไวยากรณ์ของภาษาที่ $\Sigma = \{+, a\}$ กำหนดมาให้ดังนี้

$$S \to S + S \mid a$$

จะเห็นว่าสายอักขระ $a + a + a$ สามารถทำการแปลงตัวแปรจากทางซ้ายสุดก่อนได้ 2 แบบ ดังนี้

แบบที่ 1 $S \Rightarrow S + S \Rightarrow a + S \Rightarrow a + S + S \Rightarrow a + a + S \Rightarrow a + a + a$

แบบที่ 2 $S \Rightarrow S + S \Rightarrow S + S + S \Rightarrow a + S + S \Rightarrow a + a + S \Rightarrow a + a + a$

ดังนั้น ไวยากรณ์จากตัวอย่างดังกล่าวเป็น ไวยากรณ์ที่มีความกำกวม

2.1.6 การแจงส่วน (parsing)

การแจงส่วน คือ การหาการแปลงของไวยากรณ์ G จากตัวแปรเริ่มต้น S ไปยังสายอักขระ ซึ่งถ้าไม่สามารถหาการแปลงจากตัวแปรเริ่มต้นของไวยากรณ์ไปยังสายอักขระได้ แสดงว่าสายอักขระนั้นเป็นสมาชิกของ $L(G)$ ดังนั้น การแจงส่วนจะเป็นการการตรวจสอบความเป็นสมาชิกในภาษา $L(G)$ ของสายอักขระใดๆ การแจงส่วนแบ่งมี 2 ประเภท คือ

2.1.6.1 การแจงส่วนแบบล่างขึ้นบน (bottom-up parsing)

การแจงส่วนแบบล่างขึ้นบนจะเริ่มตรวจสอบสายอักขระที่รับเข้ามากับไวยากรณ์ที่สร้างขึ้นในระดับไวยากรณ์ไม่พึ่งบริบทจากการดูสายอักขระไล่ไปหาตัวแปรเริ่มต้น S ที่ครอบคลุมสายอักขระนั้น การแจงส่วนแบบนี้ต้องอาศัยตัวแจงส่วนตาราง (chart parser) อัลกอริทึมที่นิยมคือ อัลกอริทึมซีวายเค (CYK algorithm) [6] เป็นการเขียนโปรแกรมเชิงพลวัต (dynamic programming) ซึ่งอัลกอริทึมนี้มีข้อจำกัดคือ จะรองรับเฉพาะไวยากรณ์ไม่พึ่งบริบทที่อยู่ในรูปปกติชอมสกี (Chomsky normal form) เท่านั้น

2.1.6.2 การแจงส่วนแบบบนลงล่าง (top-down parsing)

การแจงส่วนแบบบนลงล่างจะเริ่มตรวจสอบสายอักขระที่รับเข้ามากับไวยากรณ์ที่สร้างขึ้นในระดับไวยากรณ์ไม่พึ่งบริบทจากการดูตัวแปรเริ่มต้นที่ S ไล่ไปจนกระทั่งครอบคลุมสายอักขระนั้น อัลกอริทึมที่นิยม คือ อัลกอริทึมเอียร์เลย์ (Earley algorithm) [1] เป็นการเขียนโปรแกรมเชิงพลวัต ซึ่งหลักการทำงานของตัวแจงส่วนจะทำการวนซ้ำตัวกระทำการ 3 ขั้นตอนเหล่านี้ได้แก่ การทำนาย (prediction) การกราดตรวจ (scanning) และการทำให้บริบูรณ์ (completion)

2.2 การเรียนรู้เชิงอุปนัย (inductive learning)

การเรียนรู้เชิงอุปนัย หมายถึง การเรียนรู้ข้อมูลจากชุดตัวอย่างเพื่อให้ได้มาซึ่งกฎเกณฑ์ทั่วไปสำหรับอธิบายชุดตัวอย่างนั้นได้ [6] ตัวอย่างการเรียนรู้จากชุดตัวอย่าง เช่น $aaabb, aabb, aabbbb, aaabbbb$ ซึ่งอาจจะสรุปได้ว่า กฎเกณฑ์ที่ใช้อธิบายภาษานี้คือ สายอักขระที่ขึ้นต้นด้วย a ไม่จำกัดจำนวน แล้วลงท้ายด้วย b ไม่จำกัดจำนวน ซึ่งคำจำกัดความของการเรียนรู้ด้วยการอุปนัยนี้อาจจะไม่เหมือนการเรียนรู้ในความหมายทั่วๆไปตรงที่ การเรียนรู้ด้วยการอุปนัยจะสนใจแต่กรณีของการได้มาซึ่งกฎเกณฑ์ทั่วไปของข้อมูล โดยที่ยังไม่ได้คำนึงถึงคำตอบว่าสิ่งใดอยู่หรือไม่อยู่ในกลุ่มข้อมูลตัวอย่าง แต่การเรียนรู้ในความหมายทั่วไปจะสนใจในแง่ของการตอบคำถามว่าใช่หรือไม่ มากกว่ากฎเกณฑ์ที่อธิบายสภาพของข้อมูล ในการเรียนรู้เชิงอุปนัยนั้นจะระบุปัญหาของการเรียนรู้ดังนี้

1. ชั้นของฟังก์ชันหรือชั้นของภาษาที่พิจารณา เช่น กฎเกณฑ์ที่ได้จากการเรียนรู้ภาษาในระดับไวยากรณ์สม่ำเสมอ

2. สมมติฐานที่เป็นไปได้ (hypothesis space) หมายถึง ขอบเขตของสมมติฐานที่ได้จากการเรียนรู้ที่เป็นไปได้ ตัวอย่างเช่น ในการเรียนรู้ภาษาสม่ำเสมอ จะออกแบบกฎให้อยู่ในรูปออโตมาตา หรือไวยากรณ์สม่ำเสมอ ต้องคำนึงถึงสมมติฐานที่เป็นไปได้ทั้งหมด

3. ตัวอย่างที่ยอมรับได้ในการแสดงออก (admissible presentation) ตัวอย่างเช่น ข้อมูลตัวอย่างที่อยู่ในภาษา และข้อมูลตัวอย่างที่ไม่อยู่ในภาษา

4. วิธีการพิจารณาว่าสมมติฐานใดที่เป็นไปได้ ตัวอย่างเช่น อัลกอริทึม

5. ขอบเขตความสำเร็จของในการเรียนรู้ (criterion of success) เช่น การยอมรับว่าการเรียนรู้นั้นสำเร็จเมื่อพบว่า สามารถระบุหรือแยกแยะตัวอย่างที่อยู่และไม่อยู่ในภาษาได้

2.3 การอนุมานไวยากรณ์ไม่พึ่งบริบท (context-free grammar inference)

การอนุมานไวยากรณ์ไม่พึ่งบริบท คือ ปัญหาการหาไวยากรณ์ไม่พึ่งบริบท ที่สามารถระบุภาษาไม่พึ่งบริบทจากกลุ่มข้อมูลตัวอย่างที่เป็นสายอักขระในภาษา ซึ่งอาจประกอบไปด้วยกลุ่มข้อมูลที่อยู่ในภาษาที่เรียกว่า ตัวอย่างบวก และ/หรือ กลุ่มข้อมูลที่ไม่อยู่ในภาษาที่เรียกว่า ตัวอย่างลบ และการระบุความสำเร็จในการอนุมานไวยากรณ์ที่นิยมนำมาเป็นตัววัดความสำเร็จ คือ หลักการจำแนกภาษาภายในจำกัด (language identification in the limit) ของโกลด์ [2]

2.3.1 หลักการจำแนกภาษาภายในจำกัด

ตัวอย่างมุมมองการเรียนรู้ด้วยหลักการจำแนกภาษาภายในจำกัด จากเกมเดาตัวเลขที่ต่อจากลำดับ $1,3,5,\ldots$ ควรจะเป็นอะไร จากการเรียนรู้ด้วย $1,3,5,\ldots$ อาจจะเดาสมมติฐานแรกว่าจะต้องเป็นชุดเลขคี่ หรือ $2n+1$ดังนั้นเมื่อตัวต่2อไปมาเป็นเลข 7 เข้ามาสรุปว่าฟังก์ชันเป็นเลขคี่ยังคงถูกต้องอยู่ จากนั้นเมื่อตัวเลขถัดไปที่เข้ามาเป็น 11 ปรากฏว่าฟังก์ชันเลขคี่ไม่ถูกต้อง จึงจำเป็นต้องเปลี่ยนเป็นสมมติฐานด้วยการเดาว่าเป็นเลขจำนวนเฉพาะ ผลปรากฏว่าเลขที่เข้ามาใหม่ทั้งหมดเป็นลำดับดังนี้ $13,17,19,23,\ldots$ ซึ่งถูกต้องทั้งหมด ดังนั้นจึงกล่าวได้ว่า จากตัวเลขที่เข้ามา มีฟังก์ชันคือ ฟังก์ชันเลขจำนวนเฉพาะ สมมติฐานการเรียนรู้ถูกต้องตราบใดที่ยังไม่มีชุดตัวเลขที่ทำให้สมมติฐานไม่เป็นจริง การเรียนรู้ด้วยหลักการจำแนกภาษาภายในจำกัด หมายถึง การกำหนดขอบเขตความสำเร็จของการเรียนรู้ด้วยหลักการที่ว่าเมื่อมีการเรียนรู้ไปได้ระยะหนึ่งสมมติฐานถูกต้องนั้นจะไม่เปลี่ยนแปลง และจำนวนครั้งในการเปลี่ยนสมมติฐานนั้นรับประกันได้ว่ามีจำนวนจำกัด กำหนดให้ G_t แทนฟังก์ชันที่ได้จากการเรียนรู้เมื่อรับสายอักขระลำดับที่ t จากกลุ่มข้อมูลตัวอย่างแล้ว เรายอมรับว่าการเรียนรู้สำเร็จ ณ ลำดับที่ t เมื่อพบว่า

$$G_t = G_{t+1} = G_{t+2} = G_{t+3} = \ldots$$

ลำพังการเรียนรู้ด้วยหลักการจำแนกภาษาภายในจำกัดนั้นยังขาดประสิทธิภาพ จึงได้มีการนำเสนอหลักเกณฑ์เพื่อรับประกันประสิทธิภาพ โดยกล่าวว่า การเรียนรู้ใดๆนั้นนอกจากจะสามารถจำแนกเอกลักษณ์ได้แล้ว ยังต้องใช้เวลาไม่เกินฟังก์ชันพหุนาม รวมถึงอัตราการเจริญเติบโตของข้อมูลที่ได้จากเรียนรู้จะต้องไม่เกินฟังชันพหุนาม ซึ่งการอนุมานภาษาสม่ำเสมอ ประสบผลสำเร็จในหลักการจำแนกภาษาภายในจำกัด ใช้ความซับซ้อนเชิงเวลาระดับพหุนาม ด้วยอัลกอริทึมอาร์พีเอ็นไอ (Regular Positive and Negative Inference algorithm : RPNI algorithm) [8] ซึ่งได้ออโตมาตาจำกัดสถานะเชิงกำหนดที่ยอมรับทุกข้อมูลตัวอย่างบวกและปฏิเสธทุกตัวอย่างลบ ส่วนการอนุมานไวยากรณ์ไม่พึ่งบริบท ในทางทฤษฎีแล้วไม่สามารถใช้หลักการจำแนกภาษาภายในจำกัดโดยใช้ความซับซ้อนเชิงเวลาระดับพหุนามได้ [7] แต่ใน

ความรู้ปัจจุบันพบว่า ในงานวิจัยของวุฒิ สุนทรภัณฑ์ [4] ได้นำเสนออัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบทที่ใช้หลักการจำแนกภาษาภายในจำกัดโดยใช้ความซับซ้อนเชิงเวลาระดับพหุนามได้ ซึ่งรายละเอียดของงานวิจัยของวุฒิ สุนทรภัณฑ์ จะอยู่ในหัวข้องานวิจัยที่เกี่ยวข้อง

2.3.2 การเรียนรู้ด้วยการประมาณความถูกต้องโดยความน่าจะเป็น (Probably Approximately Correct learning : PAC learning)

นอกจากหลักการจำแนกโดยจำกัดแล้ว ยังมีรูปแบบการวัดความสำเร็จของการเรียนรู้อีกหลายอย่าง เช่น การเรียนรู้ด้วยการประมาณความถูกต้องโดยความน่าจะเป็น [12] เป็นการเรียนรู้โดยการคำนวณเพื่อหาความน่าจะเป็นให้ตรงตามข้อมูลตัวอย่าง โดนมีหลักเกณฑ์ว่าสามารถยอมรับค่าความน่าจะเป็นที่มีความคลาดเคลื่อนได้ระดับหนึ่งได้ ถ้าความคลาดเคลื่อนไม่เกินค่าใดค่าหนึ่ง ซึ่งค่านี้อาจจะถูกกำหนดไว้ล่วงหน้า และเมื่อใดที่ก็ตามที่ดำเนินการเรียนรู้จนได้ความคลาดเคลื่อนไม่มากไปกว่าที่กำหนดแล้ว เราสามารถยอมรับได้ว่าเรียนรู้สำเร็จ นอกจากนี้ยังมีข้อจำกัดคือ เวลาในการเรียนรู้และขนาดแบบจำลองจะต้องไม่เติบโตเกินฟังก์ชันพหุนาม จากความรู้ในปัจจุบันยังไม่มีข้อที่พิสูจน์ได้ว่า เมื่อนำมาประยุกต์ใช้กับการเรียนรู้ไวยากรณ์แล้ว จะสามารถทำนายได้ถูกต้องตามหลักการหรือไม่ มีแต่แนวโน้มว่าไม่สามารถที่จะนำมาใช้ได้เนื่องจาก มีการพิสูจน์ว่าปัญหาในการทำนายข้อมูลมีความยากเท่ากับบางปัญหาในการเข้ารหัส (cryptographic problem) [9]

2.4 งานวิจัยที่เกี่ยวข้อง

2.4.1 งานวิจัย Incremental learning of context-free grammars based on bottom-up parsing and search ของนากามุระและมัตซุโมโต

ในปี 2005 นากามุระและมัตซุโมโต [3] ได้นำเสนออัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบท โดยใช้การพิจารณาเซตแบบลำดับ (ordered set) ของข้อมูลตัวอย่างบวกและข้อมูลตัวอย่างลบ การทำงานอาศัยอัลกอริทึมการแจงส่วนซีวายเค ซึ่งเป็นเครื่องมือสำคัญในการพิจารณาการสร้างกฎของไวยากรณ์ไม่พึ่งบริบทที่อยู่ในรูปปกติชอมสกี แต่มีข้อจำกัดคือ ไม่สามารถอนุมานไวยากรณ์ไม่พึ่งบริบทที่มีตัวแปรมากกว่า 12 ตัวแปรได้ เนื่องจากติดปัญหาทางด้านเวลาที่ใช้ในการคำนวณ ซึ่งรายละเอียดของอัลกอริทึมแสดงได้ดังนี้

หลักการทำงานของอัลกอริทึมใช้การหาเซตของกฎจากตัวแปรที่สามารถแจงส่วนตัวอย่างบวกได้ แต่ไม่สามารถแจงส่วนตัวอย่างลบได้ โดยอาศัยการกำหนดตัวแปรจากตารางการแจงส่วนของซีวายเค โดยเริ่มจากตัวแปรเริ่มต้น S เพียงหนึ่งตัว เมื่อไม่สามารถสร้างกฎโดยใช้จำนวนตัวแปรที่มีอยู่ได้ จึงเพิ่มจำนวนตัวแปรขึ้นทีละตัว และกฎที่ได้จะอยู่ในบรรทัดฐานชอมสกี ส่วนเวลาการทำงานของอัลกอริทึมจะเพิ่มขึ้นเป็นแบบชี้กำลังตามจำนวนตัวแปรที่

เพิ่มขึ้น แต่จะมีข้อกำหนดว่าจะไม่ทำการสร้างกฎมากเกินกว่าค่า $K_{\max}$ เมื่อได้กฎที่สามารถสร้างตัวอย่างบวกได้หนึ่งตัวแล้ว จะนำกฎที่ได้ไปตรวจสอบกับตัวอย่างลบทั้งหมด ถ้าตรวจสอบแล้วพบว่ากฎนั้นสามารถสร้างตัวอย่างลบได้ จะทำการย้อนรอย (backtracking) ไปยังจุดทางเลือก (choice point) ซึ่งการทำงานจะวนรอบจนครบตัวอย่างบวก

อัลกอริทึมนี้ใช้ความซับซ้อนเชิงเวลาเป็นระดับเลขชี้กำลัง โดยจะขึ้นกับจำนวนกฎที่เป็นไปได้ในการสร้างแต่ละรอบของตัวอย่างบวกหนึ่งตัวเท่ากับ $O(|N|^{3K})$ เมื่อ $|N|$ เป็นจำนวนตัวแปรในขณะนั้น และ K เป็นจำนวนกฎที่เป็นไปได้ในการสร้างขึ้นใหม่ในแต่ละรอบซึ่งแปรตามการจับคู่ของตัวแปรทั้งหมดในระบบ ซึ่งจะทำให้ความซับซ้อนเชิงเวลาในงานวิจัยดังกล่าวเป็นเลขชี้กำลัง และถ้ากำหนด K ให้มีค่าต่ำจะทำให้ไม่สามารถหาไวยากรณ์ที่ได้ถูกต้อง

2.4.2 งานวิจัย Learning context-free grammars with a simplicity bias ของแลงเลย์และสโตรมส์เตน

ในปี 2000 แลงเลย์ และสโตรมส์เตน [5] ได้นำเสนออัลกอริทึมไวยากรณ์ไม่พึ่งบริบท โดยพิจารณาจากตัวอย่างบวกเพียงอย่างเดียว และไวยากรณ์ที่ใช้ทดสอบจะเป็นไวยากรณ์ที่ใช้ในภาษาอังกฤษ ซึ่งจะมีตัวดำเนินการอยู่สองแบบ คือ การสร้างตัวแปรใหม่ และการรวมตัวแปรเข้าด้วยกัน การสร้างตัวแปรตัวใหม่สร้างจากการหากลุ่มของตัวแปรจากกฎทางขวามือที่เกิดขึ้นบ่อยๆ รูปแบบการพิจารณาการสร้างตัวแปรใหม่ทำได้ดังนี้

กำหนดให้มีกฎเริ่มต้น ดังนี้

$$NP \rightarrow ART\ ADJ\ NOUN$$

$$NP \rightarrow ART\ ADJ\ ADJ\ NOUN$$

สามารถสร้างตัวแปรใหม่ $AP1$ และแปลงกฎเป็น

$$NP \rightarrow ART\ AP1$$

$$NP \rightarrow ART\ ADJ\ AP1$$

$$AP1 \rightarrow ADJ\ NOUN$$

ส่วนรูปแบบการพิจารณาการรวมตัวแปร จะทำการพิจารณาดังนี้

กำหนดให้มีกฎ

$$NP \rightarrow ART\ AP1$$

$$NP \rightarrow ART\ AP2$$

$$AP1 \rightarrow ADJ\ NOUN$$

$$AP2 \rightarrow ADJ\ AP1$$

สามารถรวมตัวแปร $AP1$ และ $AP2$ ได้เป็น

$$NP \rightarrow ART\ AP1$$

$$AP1 \rightarrow ADJ\ NOUN$$

$$AP1 \rightarrow ADJ\ AP1$$

ซึ่งอัลกอริทึมนี้จะมีตัววัดว่าสมควรจะรวมตัวแปรหรือสร้างตัวแปรใดก่อนจากหลักการของฟังก์ชันการบรรยายด้วยความยาวสั้นที่สุด (Minimal Description Length : MDL) หลักการทำงานของอัลกอริทึมจะเริ่มจากการรวมตัวแปรที่ดีที่สุดโดยวัดจากฟังก์ชันเอ็มดีแอล จนไม่สามารถรวมตัวแปรได้แล้วจึงเปลี่ยนไปสร้างตัวแปรที่ดีที่สุด โดยวัดจากฟังก์ชันเอ็มดีแอลเช่นกัน เมื่อไม่สามารถสร้างตัวแปรใหม่ได้แล้วจะเปลี่ยนไปทำการรวมตัวแปรอีกครั้ง ทำไปจนกว่าไม่สามารถหาไวยากรณ์ที่ดีขึ้นได้แล้ว จึงจบการทำงาน ซึ่งข้อเสียของอัลกอริทึมนี้คือ การใช้ตัวอย่างบวกเพียงอย่างเดียวทำให้ไวยากรณ์ที่ได้มีความกว้างมากเกินไป

2.4.3 งานวิจัย Ga-based learning of context-free grammars using tabular representations ของซากากิบารา

ในปี 2005 ซากากิบารา [10,11] ได้นำเสนออัลกอริทึมในการอนุมานไวยากรณ์ไม่พึ่งบริบทใช้การพิจารณาจากตัวอย่างบวกและลบ และใช้อัลกอริทึมเชิงพันธุกรรมเข้ามาค้นหาเซตของตัวแปรที่เป็นไปได้ โดยหาจากโครงสร้างของต้นไม้ที่ได้จากตารางแจงส่วนแบบซีวายเค ซึ่งจะเห็นว่าคล้ายกับงานวิจัยของนากามุระและมัตซุโมโต แต่การทำงานมีความไม่แน่นอนเนื่องจากใช้หลักการสุ่มในการหาประชากรใหม่ ทำให้ไวยากรณ์ที่ได้จากอัลกอริทึมในแต่ละครั้งไม่เหมือนกัน

2.4.4 งานวิจัยการปรับปรุงและพัฒนาอัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบทของนายวุฒิ สุนทรภัณฑ์

ในปี 2006 วุฒิ สุนทรภัณฑ์ [4] ได้นำเสนอการอนุมานไวยากรณ์ไม่พึ่งบริบทสำหรับภาษาไม่พึ่งบริบทรวมทั้งภาษาสม่ำเสมอ ที่มีความซับซ้อนเชิงเวลาไม่เกินฟังก์ชันพหุนาม ซึ่งหลักการทำงานของอัลกอริทึมจะพิจารณาสร้างกฎวนซ้ำจากข้อมูลตัวอย่างบวก และเพื่อไม่ให้ไวยากรณ์กว้างจนเกินไป จะใช้ตัวอย่างลบมาร่วมพิจารณา

การทำงานของอัลกอริทึมเริ่มจากการรับตัวอย่างบวกมาทีละตัวอย่าง โดยที่แต่ละรอบจะเริ่มจากการตรวจสอบว่า ข้อมูลตัวอย่างบวกที่รับเข้ามาสามารถหาการแปลงจากตัวแปรเริ่มต้นจากกฎที่มีอยู่ก่อนหน้าได้หรือไม่ ถ้าสามารถหาการแปลงของข้อมูลตัวอย่างบวกได้ แสดงว่าไม่ต้องทำการเพิ่มกฎหรือเปลี่ยนกฎแต่อย่างใด แต่ถ้าไม่สามารถหาการแปลงของข้อมูลตัวอย่างบวกได้ จะเข้าสู่การทำงาน 3 ขั้นตอน ดังนี้

1. ขั้นตอนการแทนที่สายอักขระของข้อมูลตัวอย่างบวกที่สามารถแปลงเป็นตัวแปรที่มีอยู่ก่อนหน้านี้

2. ขั้นตอนการสร้างกฎใหม่ โดยเปรียบเทียบตัวอย่างบวกที่ถูกแทนด้วยตัวแปร กับสายอักขระของกฎทางขวามือทุกกฎที่กฎทางซ้ายมือเป็นตัวแปรเริ่มต้น S เพื่อหาสายอักขระย่อยที่มีความยาวมากที่สุดที่เกิดขึ้น แล้วนำเอาสายอักขระของตัวอย่างบวกที่พิจารณาอยู่ และสายอักขระที่เปรียบเทียบ มาทำการหาสายอักขระที่เหลือจากการแทนสายอักขระย่อยที่ยาวที่สุด แล้วนำส่วนที่เหลือจากการแทนสายอักขระย่อยมาเปรียบเทียบเพื่อสร้างเป็นกฎวนซ้ำ แล้วจึงนำกฎใหม่ที่สร้างไปรวมกับกฎที่มีอยู่แล้ว ได้เป็นไวยากรณ์สมมติฐาน จากนั้นนำไวยากรณ์สมมติฐานไปทดสอบการแปลงของข้อมูลตัวอย่างลบ ถ้าพบว่ามีข้อมูลตัวอย่างลบที่สามารถแปลงได้จากไวยากรณ์สมมติฐาน แสดงว่าไม่ยอมรับไวยากรณ์สมมติฐานนั้น ถ้าพบว่าข้อมูลตัวอย่างลบทุกตัวไม่สามารถแปลงได้จากไวยากรณ์สมมติฐาน แสดงว่ายอมรับไวยากรณ์สมมติฐานนั้น จะกำหนดไวยากรณ์สมมติฐานให้เป็นไวยากรณ์ที่ถูกต้อง

3. ทำการยุบรวมตัวแปรที่สามารถรวมกันได้ จะได้ไวยากรณ์ใหม่ ถือเป็นการสิ้นสุดรอบการทำงานจากการเรียนรู้จากข้อมูลตัวอย่างบวก 1 ตัว

ทำการวนรอบตรวจสอบข้อมูลตัวอย่างบวกทีละตัวจนครบทุกตัวอย่างจะได้ไวยากรณ์ไม่พึ่งบริบทสุดท้ายที่ถูกต้อง ซึ่งอัลกอริทึมใช้ความซับซ้อนเชิงเวลาเป็น $O(\| S_P \|^2 \| S_N \|^3)$ เมื่อ $\| S_P \|$ เป็นผลรวมของความยาวสายอักขระทุกตัวที่เป็นสมาชิกของเซตตัวอย่างบวก และ $\| S_N \|$ เป็นผลรวมของความยาวสายอักขระทุกตัวที่เป็นสมาชิกของเซตตัวอย่างลบ

บทที่ 3

อัลกอริทึมสำหรับสร้างไวยากรณ์ไม่พึ่งบริบทแบบเชื่อมตรงโดยใช้ลำดับของกฎแฝง

ในงานวิจัยนี้เสนอการสร้างไวยากรณ์ไม่พึ่งบริบทแบบใหม่ โดยอาศัยการเรียนรู้รูปแบบลำดับของกฎแฝงด้วยไวยากรณ์สม่ำเสมอ สำหรับรายละเอียดในบทนี้จะเริ่มต้นกล่าวถึงการวิเคราะห์รูปแบบของปัญหา การทำงานของอัลกอริทึม อัลกอริทึมสำหรับสร้างไวยากรณ์ไม่พึ่งบริบทโดยใช้ลำดับของกฎแฝง ผลการวิเคราะห์ สรุป และเพื่อให้ง่ายต่อความเข้าใจการทำงานในแต่ละขั้นตอน จะมีการยกตัวอย่างประกอบการทำงานซึ่งสามารถช่วยให้เห็นการทำงานในแต่ละขั้นตอนได้ชัดเจนยิ่งขึ้น

3.1 การวิเคราะห์รูปแบบของปัญหา

ทั้งนี้เนื่องจากภาษารูปนัยตามทฤษฎีของภาษานั้น ความซับซ้อนของภาษาสามารถแบ่งออกได้เป็นหลายระดับชั้น โดยความซับซ้อนเหล่านั้นจะแสดงให้เห็นได้จากคุณสมบัติของไวยากรณ์ที่ใช้ในการอธิบายภาษา งานวิจัยนี้มีขอบเขตของการเรียนรู้ที่มุ่งเน้นไปที่ภาษาที่มีความซับซ้อนไม่เกินไปกว่าภาษาไม่พึ่งบริบท

ข้อมูลตัวอย่างที่นำมาใช้ในการเรียนรู้ ต้องผ่านกระบวนการแจงส่วนแบบเอียร์เลย์เพื่อดูลำดับของกฎไวยากรณ์ที่ถูกเลือกใช้ตามลำดับก่อนหลัง ซึ่งเป็นข้อมูลสำคัญสำหรับเรียนรู้รูปแบบลำดับของการใช้กฎที่เกิดขึ้นภายในตัวอย่างแต่ละตัวอย่าง ผลลัพธ์ที่ได้คือ ภาษาแสดงลำดับของการใช้กฎของไวยากรณ์ไม่พึ่งบริบทที่ถูกเลือกใช้ สามารถอธิบายได้ด้วยไวยากรณ์สม่ำเสมอ โดยพิจารณาการเกิดซ้ำของรูปแบบไปพร้อมกัน

เกณฑ์ที่ใช้ในการประเมินและวัดประสิทธิภาพของอัลกอริทึมคือ แบบจำลองสามารถตอบความเป็นสมาชิกของข้อมูลตัวอย่างได้อย่างถูกต้อง รวมทั้งพิจารณาปริมาณตัวอย่างที่ใช้ในการลู่เข้าของการอนุมานภาษา

จากที่กล่าวมานี้สามารถกำหนดปัญหาของงานวิจัยได้เป็นข้อสรุปดังต่อไปนี้

ปัญหาของงานวิจัย กำหนดให้

$L(G)$	ภาษาไม่พึ่งบริบทเริ่มต้น เมื่อ $G = (V, \Sigma, P, S)$
Σ	เป็นเซตจำกัดของอักขระ
$a, b, ...$	เป็นอักขระที่เป็นสมาชิกของเซต Σ โดยอักขระมีจำนวนรูปแบบที่แตกต่างกันเป็นจำนวนจำกัด
P	เป็นเซตจำกัดของกฎไวยากรณ์ไม่พึ่งบริบท
$r_1, r_2, r_3, ...$	เป็นกฎที่เป็นสมาชิกของเซต P โดยกฎจะมีจำนวนรูปแบบที่แตกต่างกันเป็นจำนวนจำกัด

S_P เป็นเซตจำกัดของข้อมูลตัวอย่างบวก โดยที่ตัวอย่างเหล่านี้จะต้องมีความยาวที่จำกัด และอักขระทั้งหมดต้องมาจากเซตอักขระ Σ

S_N เป็นเซตจำกัดของข้อมูลตัวอย่างลบ ใช้สำหรับทดสอบความถูกต้อง โดยตัวอย่างเหล่านี้จะต้องมีความยาวที่จำกัด และอักขระทั้งหมดต้องมาจากเซตอักขระ Σ

S_{PT} เป็นเซตจำกัดของข้อมูลสายลำดับของกฎแฝงที่ได้มาจากการแจงส่วนแบบเอียร์เลย์ด้วยตัวอย่างบวก โดยข้อมูลเหล่านี้จะต้องมีลำดับของกฎแฝงที่มีลำดับที่จำกัด และสายลำดับทั้งหมดต้องมาจากเซตกฎ P

S_{NT} เป็นเซตจำกัดของข้อมูลสายลำดับของกฎแฝงที่ได้มาจากการแจงส่วนแบบเอียร์เลย์ด้วยตัวอย่างลบ โดยข้อมูลเหล่านี้จะต้องมีลำดับของการใช้กฎที่มีลำดับที่จำกัด และสายลำดับทั้งหมดต้องมาจากเซตกฎ P

Σ_T เป็นเซตจำกัดของกฎไวยากรณ์ $\Sigma_T \in P$

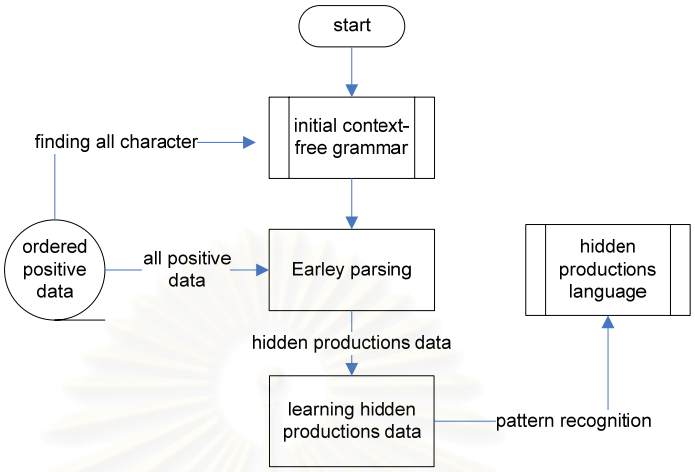
ในการอนุมานภาษา L_T ที่เป็นภาษาแสดงลำดับของการใช้กฎจากไวยากรณ์ไม่พึ่งบริบทใด ๆจาก $L(G)$ โดยที่ L_T จะต้องมีคุณสมบัติ ดังนี้

$$S_{PT} \subseteq L_T \text{ และ } S_{NT} \cap L_T = \phi$$

3.2 การทำงานของอัลกอริทึม

การทำงานของอัลกอริทึม สำหรับสร้างไวยากรณ์ไม่พึ่งบริบทโดยใช้ลำดับของกฎแฝงในงานวิจัยนี้ มีข้อกำหนดเบื้องต้นคือ เริ่มต้นพิจารณาข้อมูลตัวอย่างบวกทั้งหมด S_P เพื่อพิจารณาจำนวนอักขระทั้งหมดที่ต้องใช้ นำมาสร้างเป็นเซตจำกัดของอักขระ Σ จากนั้นเริ่มสร้างไวยากรณ์ไม่พึ่งบริบทเริ่มต้น $L(G)$ ที่กว้างครอบคลุมข้อมูลตัวอย่างบวก S_P ทั้งหมด ซึ่งการสร้างไวยากรณ์เริ่มต้นจะจะกล่าวในส่วนถัดไป หลังจากนั้นพิจารณาข้อมูลตัวอย่างบวกทีละตัวอย่าง แล้วใช้การแจงส่วนแบบเอียร์เลย์เพื่อดึงข้อมูลสายลำดับกฎของไวยากรณ์ไม่พึ่งบริบทที่ถูกใช้งานจากตัวอย่างบวกที่กำลังพิจารณาอยู่ ได้เป็นสมาชิกในเซต S_{PT} จากนั้นนำข้อมูลลำดับการใช้กฎในเซตดังกล่าวมาสร้างเป็นภาษาใหม่สำหรับอธิบายรูปแบบลำดับกฎที่เกิดขึ้น ซึ่งภาษาดังกล่าวสามารถอธิบายได้ด้วยไวยากรณ์สม่ำเสมอ โดยในแต่ละรอบของการพิจารณาทีละตัวอย่างบวก จะทำการผสานและลดรูปภาษาสม่ำเสมอที่ได้มาจากรอบก่อนหน้า ซึ่งผลลัพธ์สุดท้ายที่ได้จากอัลกอริทึม คือ L_T หรือภาษาสม่ำเสมอที่ใช้อธิบายข้อมูลลำดับของการใช้กฎไวยากรณ์ไม่พึ่งบริบทที่ครอบคลุมข้อมูลตัวอย่างบวกทั้งหมด ส่วนข้อมูลตัวอย่างลบหรือเซต

S_N ใช้สำหรับทดสอบความถูกต้องของ L_T โดยการทำงานของอัลกอริทึมสามารถอธิบายได้ด้วยผังงาน ดังรูปที่ 3.1



รูปที่ 3.1 ผังงานการสร้างไวยากรณ์ไม่พึ่งบริบทแบบเชื่อมตรงโดยใช้ลำดับของกฎแฝง

3.3 อัลกอริทึมสำหรับสร้างไวยากรณ์โดยใช้ลำดับของกฎแฝง

การทำงานของอัลกอริทึมเริ่มจากการพิจารณาจำนวนอักขระที่ต้องใช้ พร้อมทำการรับตัวอย่างบวกเข้ามาทีละตัว สามารถแบ่งการทำงานเป็น 3 ขั้นตอนหลักๆ ดังนี้

1. อัลกอริทึมสำหรับสร้างไวยากรณ์ไม่พึ่งบริบทเริ่มต้น
2. อัลกอริทึมสำหรับดึงลำดับการใช้กฎแฝงจากการแจงส่วนแบบเอียร์เลย์
3. อัลกอริทึมสำหรับสร้างไวยากรณ์อธิบายข้อมูลสายลำดับของกฎแฝง

ซึ่งขั้นตอนการทำงานหลักๆ ของอัลกอริทึมโดยรวมมีการทำงานดังนี้

Algorithm

Input: an ordered set S_P of positive sample strings.

Output : language L_T such that all the strings in S_P derived from L_T

L = CFGGenerating(S_P)

S_{PT} = EarleyParsing(L, S_P)

begin

L_T = null

for I =1 to $|S_{PT}|$

L_T = Merging & Reduction(S_{PT}[i], L_T)

end for

end

จากอัลกอริทึมดังกล่าว CFGGenerating(S_P) คือ อัลกอริทึมการสร้างไวยากรณ์ไม่พึ่งบริบทเริ่มต้น, EarleyParsing(L,S_P) อัลกอริทึมการดึงลำดับการใช้กฎแฝงจากการแจงส่วนแบบเอียร์เลย์ และ Merging & Reduction(S_{PT}[i],L_T) คือ อัลกอริทึมการสร้างไวยากรณ์สำหรับอธิบายข้อมูลสายลำดับของกฎแฝง ซึ่งจะขออธิบายตามลำดับ ดังนี้

3.3.1 อัลกอริทึมสำหรับสร้างไวยากรณ์ไม่พึ่งบริบทเริ่มต้น

หลักการของอัลกอริทึมสร้างไวยากรณ์ไม่พึ่งบริบทเริ่มต้น $G = (V, \Sigma, P, S)$ พิจารณาอักขระทั้งหมดที่ใช้จากตัวอย่างบวกทั้งหมด สร้างเป็นเซตจำกัดของอักขระ Σ ทำการกำหนดเซตจำกัดของตัวแปร V จากนั้นทำการสร้างกฎไวยากรณ์ไม่พึ่งบริบทเริ่มต้นที่ครอบคลุมตัวอย่างบวกทั้งหมด หรือในอีกความหมายคือ ทุกข้อมูลตัวอย่างที่เป็นสมาชิกในภาษา Σ^* นั้นสามารถผลิตได้จาก $L(G)$

หลักการสร้างกฎไวยากรณ์เริ่มต้นมีดังนี้ กำหนดให้กฎ r_i เป็น $A \to \alpha$ *เมื่อ* $r_i \in P, A \in V$ เมื่อ α คือ รูปประโยคของกฎทางฝั่งขวา ซึ่งหากไม่มีข้อกำหนดใดๆ ในการสร้างไวยากรณ์เริ่มต้น รูปประโยคกฎทางฝั่งขวาที่เป็นไปได้จะมีรูปแบบที่ไม่จำกัด หรือในรูปแบบดังนี้

$$\alpha \in (V + \Sigma)^*$$

จากรูปแบบที่ไม่จำกัด จะเห็นว่า เกิดความเกินจำเป็นในการสร้างกฎขึ้น ดังนั้นจึงต้องกำหนดขอบเขตความยาวสูงสุดของรูปประโยคของกฎทางฝั่งขวา α ให้มีความยาวที่จำกัดที่ไม่เกิน $l_{\max}$ และจากนิยาม 2.20 ความกำกวมของไวยากรณ์สามารถเกิดขึ้นได้ ซึ่งการตัดทอนความเกินจำเป็นสามารถช่วยลดความกำกวมของไวยากรณ์ลงได้

ในส่วนของรูปประโยคของกฎทางฝั่งขวา α ต้องกำหนดให้มีจำนวนตัวแปรปนอยู่ได้ไม่เกิน 1 ตัว หรือให้เป็นอักขระว่าง เพื่อการแจงส่วนด้วยอัลกอริทึมเอียร์เลย์จะสามารถเลือกลำดับของกฎแฝงได้จากระดับความสูงของต้นไม้จากระดับบนลงมาระดับล่าง

$$\alpha \in \Sigma^* V \Sigma^* \cup \varepsilon$$

ดังนั้นจากไวยากรณ์เริ่มต้นที่ได้จากอัลกอริทึมนี้สามารถสร้างได้มากกว่า 1 แบบ ในกรณีที่ $l_{\max} = 3$ ให้เซตจำกัดของอักขระ $\Sigma = \{a, b\}$ และ $\Sigma = \{a, b, c\}$ สามารถสร้างไวยากรณ์ไม่พึ่งบริบทเริ่มต้นได้ ดังแสดงในตัวอย่างที่ 3.1 และ 3.2 ตามลำดับ

ตัวอย่างที่ 3.1 *ภาษาที่มีชุดตัวอักษร* $\Sigma = \{a, b\}$ *สร้างไวยากรณ์ไม่พึ่งบริบทเริ่มต้นได้ดังนี้*
วิธีทำ

ไวยากรณ์เริ่มต้นแบบที่ 1

กำหนดให้ $G_1 = (V_1, \Sigma, P_1, S)$

จำนวนตัวแปร $|V_1| = 2$ และ $l_{\max} = 3$

$V_1 = \{S_0, S_1\}, S = S_0$

P_1 เป็นเซตจำกัดของกฎไวยากรณ์ไม่พึ่งบริบท เป็นดังนี้

$S_0 \rightarrow aS_0a \mid aS_0b \mid bS_0a \mid bS_0b \mid S_1 \mid \varepsilon$

$S_1 \rightarrow aS_1 \mid bS_1 \mid S_0 \mid \varepsilon$

ไวยากรณ์เริ่มต้นแบบที่ 2

กำหนดให้ $G_2 = (V_2, \Sigma, P_2, S)$

จำนวนตัวแปร $|V_2| = 3$ และ $l_{\max} = 3$

$V_2 = \{S_0, S_1, S_2\}, S = S_0$

P_2 เป็นเซตจำกัดของกฎไวยากรณ์ไม่พึ่งบริบท เป็นดังนี้

$S_0 \rightarrow aS_0a \mid aS_0b \mid bS_0a \mid bS_0b \mid S_1 \mid \varepsilon$

$S_1 \rightarrow aaS_1 \mid abS_1 \mid baS_1 \mid bbS_1 \mid S_2 \mid \varepsilon$

$S_2 \rightarrow aS_2 \mid bS_2 \mid S_0 \mid \varepsilon$

จากตัวอย่างที่ 3.1 จะเห็นว่าทุกข้อมูลตัวอย่างที่เป็นสมาชิกในภาษา Σ^* สามารถผลิตได้จากไวยากรณ์เริ่มต้นแบบที่ 1 (G_1) หรือไวยากรณ์เริ่มต้นแบบที่ 2 (G_2) □

ตัวอย่างที่ 3.2 *ภาษาที่มีชุดตัวอักษร* $\Sigma = \{a, b, c\}$ *สร้างไวยากรณ์ไม่พึ่งบริบทเริ่มต้นได้ดังนี้*
วิธีทำ

ไวยากรณ์เริ่มต้นแบบที่ 1

กำหนดให้ $G_1 = (V_1, \Sigma, P_1, S)$

จำนวนตัวแปร $|V_1| = 2$ และ $l_{\max} = 3$

$V_1 = \{S_0, S_1\}, S = S_0$

P_1 เป็นเซตจำกัดของกฎไวยากรณ์ไม่พึ่งบริบท เป็นดังนี้

$S_0 \rightarrow aS_0a \mid aS_0b \mid aS_0c \mid bS_0a \mid bS_0b \mid bS_0c \mid$
$cS_0a \mid cS_0b \mid cS_0c \mid S_1 \mid \varepsilon$

$S_1 \rightarrow aS_1 \mid bS_1 \mid cS_1 \mid S_0 \mid \varepsilon$

ไวยากรณ์เริ่มต้นแบบที่ 2

กำหนดให้ $G_2 = (V_2, \Sigma, P_2, S)$

จำนวนตัวแปร $|V_2| = 3$ และ $l_{\max} = 3$

$V_2 = \{S_0, S_1, S_2\}, S = S_0$

P_2 เป็นเซตจำกัดของกฎไวยากรณ์ไม่พึ่งบริบท เป็นดังนี้

$S_0 \rightarrow aS_0a \mid aS_0b \mid aS_0c \mid bS_0a \mid bS_0b \mid bS_0c$

$cS_0a \mid cS_0b \mid cS_0c \mid S_1 \mid \varepsilon$

$S_1 \rightarrow aaS_1 \mid abS_1 \mid acS_1 \mid baS_1 \mid bbS_1 \mid bcS_1$

$caS_1 \mid cbS_1 \mid ccS_1 \mid S_2 \mid \varepsilon$

$S_2 \rightarrow aS_2 \mid bS_2 \mid cS_2 \mid S_0 \mid \varepsilon$

จากตัวอย่างที่ 3.2 จะเห็นว่าทุกข้อมูลตัวอย่างที่เป็นสมาชิกในภาษา Σ^* สามารถผลิตได้จากไวยากรณ์เริ่มต้นแบบที่ 1 (G_1) หรือไวยากรณ์เริ่มต้นแบบที่ 2 (G_2) □

ดังนั้นในงานวิจัยนี้ เราจะเริ่มต้นสร้างไวยากรณ์เริ่มต้นโดยใช้ไวยากรณ์เริ่มต้นแบบที่ 1 เนื่องจากไวยากรณ์มีขนาดเล็กกว่าเมื่อเทียบกับไวยากรณ์เริ่มต้นแบบที่ 2

3.3.2 อัลกอริทึมสำหรับดึงลำดับการใช้กฎแฝงจากการแจงส่วนแบบเอียร์เลย์

อัลกอริทึมสำหรับดึงลำดับการใช้กฎแฝง ประยุกต์ใช้อัลกอริทึมเอียร์เลย์ซึ่งเป็นการแจงส่วนแบบบนลงล่างตามที่ได้กล่าวในเนื้อหาส่วนที่ 2.1.6.2 เพื่อดูการแปลงโครงสร้างการกำเนิดของเซตตัวอย่าง S_P ทั้งหมด ทำให้ได้เซตข้อมูลสายลำดับของการใช้กฎแฝง S_{PT}

นิยามที่ 3.1 *ลำดับของกฎแฝง (sequences of hidden productions) หมายถึง กฎสองกฎที่มีการเลือกใช้ในลำดับติดกันตามการแจงส่วนแบบเอียร์เลย์ โดยในการแจงส่วนเอียร์เลย์ของสายอักขระใด ๆเพื่อหาการแปลง พบว่า ถ้ากฎ r_i ที่ถูกเลือกใช้เป็นลำดับถัดจาก r_{i-1} สามารถเขียนให้อยู่ในรูปสายลำดับของการใช้กฎ $r_{i-1}r_i$*

ตัวอย่างต่อไป เป็นตัวอย่างแสดงการดึงสายลำดับของกฎแฝงจากการแจงส่วนแบบเอียร์เลย์ ซึ่งได้เซตสายลำดับการใช้กฎแฝง S_{PT}

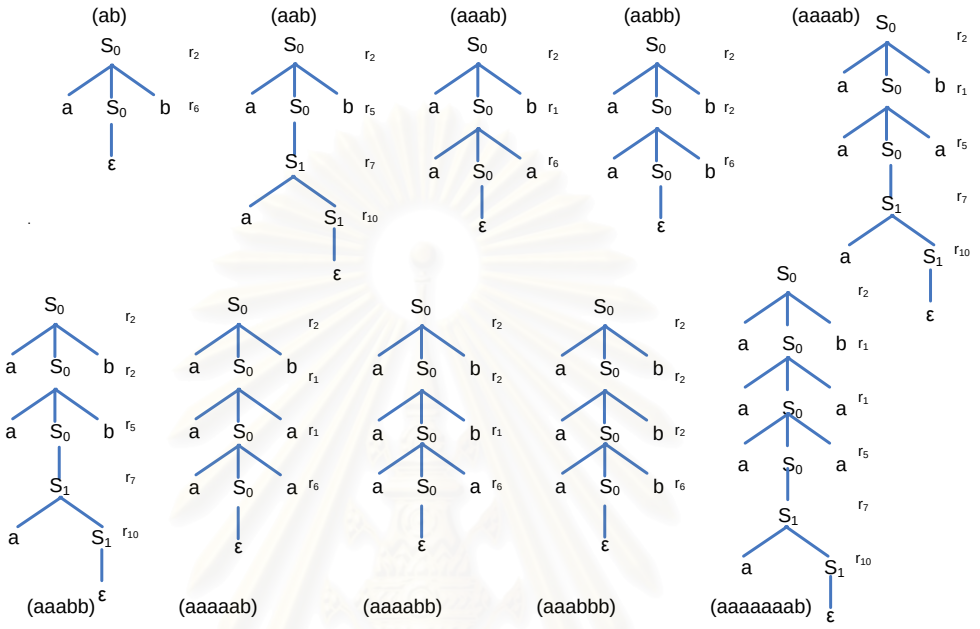
ตัวอย่างที่ 3.3 *กำหนดให้เซตข้อมูลตัวอย่างเป็นดังนี้*

$\{ab, aab, aaab, aabb, aaaab, aaabb, aaaaab, aaaabb, aaabbb, aaaaaaab,$
$aaaaabb, aaaabbb, aaaaaaab, aaaaaabb, aaaaabbb, aaaabbbb\}$

โดยให้ไวยากรณ์ไม่พึ่งบริบทเริ่มต้นเป็น $G = (V, \Sigma, P, S),\ V = \{S_0, S_1\}, S = S_0$ *และ P เป็นเซตจำกัดของกฎไวยากรณ์ ดังนี้*

$$S_0 \rightarrow aS_0a \mid aS_0b \mid bS_0a \mid bS_0b \mid S_1 \mid \varepsilon$$
$$r_1 \quad r_2 \quad r_3 \quad r_4 \quad r_5 \quad r_6$$
$$S_1 \rightarrow aS_1 \mid bS_1 \mid S_0 \mid \varepsilon$$
$$r_7 \quad r_8 \quad r_9 \quad r_{10} \quad (\text{เมื่อ } r_i \in P)$$

วิธีทำ



รูปที่ 3.2 ต้นไม้ที่ได้จากการแจงส่วนแบบเอียร์เลย์ด้วยตัวอย่างที่ 3.3

ตารางที่ 3.1 สายลำดับของกฎแฝงที่ได้จากการแจงส่วนแบบเอียร์เลย์ในตัวอย่างที่ 3.3

ลำดับที่	ข้อมูลตัวอย่าง	สายลำดับของกฎแฝง
1	ab	$r_2\ r_6$
2	aab	$r_2\ r_5\ r_7\ r_{10}$
3	aaab	$r_2\ r_1\ r_6$
4	aabb	$r_2\ r_2\ r_6$
5	aaaab	$r_2\ r_1\ r_5\ r_7\ r_{10}$
6	aaabb	$r_2\ r_2\ r_5\ r_7\ r_{10}$
7	aaaaab	$r_2\ r_1\ r_1\ r_6$
8	aaaabb	$r_2\ r_2\ r_1\ r_6$
9	aaabbb	$r_2\ r_2\ r_2\ r_6$
10	aaaaaab	$r_2\ r_1\ r_1\ r_5\ r_7\ r_{10}$
11	aaaaabb	$r_2\ r_2\ r_1\ r_5\ r_7\ r_{10}$
12	aaaabbb	$r_2\ r_2\ r_2\ r_5\ r_7\ r_{10}$

13	aaaaaaab	$r_2\ r_1\ r_1\ r_1\ r_6$
14	aaaaaabb	$r_2\ r_2\ r_1\ r_1\ r_6$
15	aaaaabbb	$r_2\ r_2\ r_2\ r_1\ r_6$
16	aaaabbbb	$r_2\ r_2\ r_2\ r_2\ r_6$

ใช้อัลกอริทึมเอียร์เลย์เพื่อแสดงลำดับการแปลงของสายอักขระจากต้นไม้การแจงส่วนดังรูปที่ 3.2 จะได้สายลำดับของกฎแฝงดังตารางที่ 3.1 □

3.3.3 อัลกอริทึมสำหรับสร้างไวยากรณ์อธิบายข้อมูลสายลำดับของกฎแฝง

หลังจากได้ข้อมูลสายลำดับของกฎแฝง S_{PT} จากขั้นตอนที่ 3.3.2 นำข้อมูลดังกล่าวเข้าสู่อัลกอริทึมการสร้างไวยากรณ์ ซึ่งอัลกอริทึมนี้จะทำการผสานและลดรูปลำดับการใช้กฎแฝงที่เกิดขึ้นด้วยการใช้การบรรยายสม่ำเสมอ โดยมีข้อกำหนดอยู่ว่า

กำหนดให้ β และ α เป็นสายลำดับของกฎแฝง โดยที่ $\beta, \alpha \in (r_i + r_{i+1} + ...)^*$ สามารถทำการผสานและลดรูปการบรรยายสม่ำเสมอได้ตามกฎดังนี้

กฎข้อที่ 1 : สายลำดับของกฎแฝง β และ α ที่เกิดขึ้นในภาษาเดียวกัน ผสานให้อยู่ในรูป $\beta + \alpha$ ได้

กฎข้อที่ 2 : สายลำดับของกฎแฝง ε, β และ $\beta\beta$ ที่เกิดขึ้นในภาษาเดียวกัน ผสานและลดรูปให้อยู่ในรูป β^* ได้

กฎข้อที่ 3 : สายลำดับของกฎแฝง β และ $\beta\beta$ ที่เกิดขึ้นในภาษาเดียวกัน ผสานและลดรูปให้อยู่ในรูป $\beta\beta^*$ ได้

กฎข้อที่ 4 : สายลำดับของกฎแฝง $\varepsilon, \beta, \alpha, \beta\beta, \beta\alpha, \alpha\beta$ และ $\alpha\alpha$ ที่เกิดขึ้นในภาษาเดียวกัน ผสานและลดรูปให้อยู่ในรูป $(\beta + \alpha)^*$ ได้

ตัวอย่างต่อไป เป็นตัวอย่างแสดงการใช้อัลกอริทึมสำหรับสร้างไวยากรณ์จากข้อมูลสายลำดับของกฎแฝงที่ได้จากตัวอย่างที่ 3.3

ตัวอย่างที่ 3.4 *ทำการสร้างไวยากรณ์สำหรับอธิบายเซตข้อมูลดังกล่าว โดยใช้เซตสายลำดับของกฎแฝงจากตัวอย่างที่ 3.3 ซึ่งมีสมาชิกทั้งหมด 16 ตัวเป็นดังนี้*

S_{PT} = { r_2r_6, $r_2r_5r_7r_{10}$, $r_2r_1r_6$, $r_2r_2r_6$, $r_2r_1r_5r_7r_{10}$, $r_2r_2r_5r_7r_{10}$, $r_2r_1r_1r_6$,
$r_2r_2r_1r_6$, $r_2r_2r_2r_6$, $r_2r_1r_1r_5r_7r_{10}$, $r_2r_2r_1r_5r_7r_{10}$, $r_2r_2r_2r_5r_7r_{10}$,
$r_2r_1r_1r_1r_6$, $r_2r_2r_1r_1r_6$, $r_2r_2r_2r_1r_6$, $r_2r_2r_2r_2r_6$}

วิธีทำ

รอบที่ 1 : จากข้อมูลสายลำดับของกฎแฝง r_2r_6 ที่ได้จากตัวอย่างบวกตัวแรก คือ ab ทำการผสานการบรรยายไวยากรณ์สม่ำเสมอเป็น

r_2r_6

รอบที่ 2 : จากข้อมูลสายลำดับของกฎแฝง $r_2r_5r_7r_{10}$ ที่ได้จากตัวอย่างบวกตัวที่ 2 คือ aab ทำการผสานการบรรยายไวยากรณ์สม่ำเสมอเป็น

$r_2r_6+r_2r_5r_7r_{10}$

รอบที่ 3 : จากข้อมูลสายลำดับของกฎแฝง $r_2r_1r_6$ ที่ได้จากตัวอย่างบวกตัวที่ 3 คือ aaab ทำการผสานการบรรยายไวยากรณ์สม่ำเสมอเป็น

$r_2r_6+r_2r_5r_7r_{10}+r_2r_1r_6$

รอบที่ 4 : จากข้อมูลสายลำดับของกฎแฝง $r_2r_2r_6$ ที่ได้จากตัวอย่างบวกตัวที่ 4 คือ aabb ทำการผสานและลดรูปการบรรยายไวยากรณ์สม่ำเสมอเป็น

$r_2r_2^*r_6+r_2r_5r_7r_{10}+r_2r_1r_6$

รอบที่ 5 : จากข้อมูลสายลำดับของกฎแฝง $r_2r_1r_5r_7r_{10}$ ที่ได้จากตัวอย่างบวกตัวที่ 5 คือ aaaab ทำการผสานการบรรยายไวยากรณ์สม่ำเสมอเป็น

$r_2r_2^*r_6+r_2r_5r_7r_{10}+r_2r_1r_6+r_2r_1r_5r_7\ r_{10}$

รอบที่ 6 : จากข้อมูลลำดับกฎ $r_2r_2r_5r_7r_{10}$ ที่ได้จากตัวอย่างบวกตัวที่ 6 คือ aaabb ทำการผสานและลดรูปการบรรยายไวยากรณ์สม่ำเสมอเป็น

$r_2r_2^*r_6+r_2r_2^*r_5r_7r_{10}+r_2r_1r_6+r_2r_1r_5r_7r_{10}$

รอบที่ 7 : จากข้อมูลสายลำดับของกฎแฝง $r_2r_1r_1r_6$ ที่ได้จากตัวอย่างบวกตัวที่ 7 คือ aaaaab ทำการผสานและลดรูปการบรรยายไวยากรณ์สม่ำเสมอเป็น

$r_2r_2^*r_6+r_2r_2^*r_5r_7r_{10}+r_2r_1^*r_6+r_2r_1r_5r_7r_{10}$

รอบที่ 8 : จากข้อมูลสายลำดับของกฎแฝง $r_2r_2r_1r_6$ ที่ได้จากตัวอย่างบวกตัวที่ 8 คือ aaaabb ทำการผสานและลดรูปการบรรยายไวยากรณ์สม่ำเสมอเป็น

$r_2r_2^*r_5r_7r_{10}+r_2r_2^*r_1^*r_6+r_2r_1r_5r_7r_{10}$

รอบที่ 9 : จากข้อมูลสายลำดับของกฎแฝง $r_2r_2r_2r_6$ ที่ได้จากตัวอย่างบวกตัวที่ 9 คือ aaabbb การบรรยายไวยากรณ์สม่ำเสมอไม่เปลี่ยนแปลง

รอบที่ 10 : จากข้อมูลสายลำดับของกฎแฝง $r_2r_1r_1r_5r_7r_{10}$ ที่ได้จากตัวอย่างบวกตัวที่ 10 คือ aaaaaab ทำการผสานและลดรูปการบรรยายไวยากรณ์สม่ำเสมอเป็น

$$r_2r_2{}^*r_5r_7r_{10}+r_2r_2{}^*r_1{}^*r_6+r_2r_1{}^*r_5r_7r_{10}$$

รอบที่ 11 : จากข้อมูลสายลำดับของกฎแฝง $r_2r_2r_1r_5r_7r_{10}$ ที่ได้จากตัวอย่างบวกตัวที่ 11 คือ aaaaabb ทำการผสานและลดรูปการบรรยายไวยากรณ์สม่ำเสมอเป็น

$$r_2r_2{}^*r_1{}^*r_6+r_2r_2{}^*r_1{}^*r_5r_7r_{10}$$

รอบที่ 12 : จากข้อมูลสายลำดับของกฎแฝง $r_2r_2r_2r_5r_7r_{10}$ ที่ได้จากตัวอย่างบวกตัวที่ 12 คือ aaaabbb การบรรยายไวยากรณ์สม่ำเสมอไม่เปลี่ยนแปลง

รอบที่ 13 : จากข้อมูลสายลำดับของกฎแฝง $r_2r_1r_1r_1r_6$ ที่ได้จากตัวอย่างบวกตัวที่ 13 คือ aaaaaaab การบรรยายไวยากรณ์สม่ำเสมอไม่เปลี่ยนแปลง

รอบที่ 14 : จากข้อมูลสายลำดับของกฎแฝง $r_2r_2r_1r_1r_6$ ที่ได้จากตัวอย่างบวกตัวที่ 14 คือ aaaaaabb การบรรยายไวยากรณ์สม่ำเสมอไม่เปลี่ยนแปลง

รอบที่ 15 : จากข้อมูลสายลำดับของกฎแฝง $r_2r_2r_2r_1r_6$ ที่ได้จากตัวอย่างบวกตัวที่ 15 คือ aaaaabbb การบรรยายไวยากรณ์สม่ำเสมอไม่เปลี่ยนแปลง

รอบที่ 16 : จากข้อมูลสายลำดับของการใช้กฎ $r_2r_2r_2r_2$ ที่ได้จากตัวอย่างบวกตัวที่ 16 คือ aaaabbbb การบรรยายไวยากรณ์สม่ำเสมอไม่เปลี่ยนแปลง

สังเกตเห็นว่าตั้งแต่สายลำดับของกฎแฝงตั้งแต่รอบที่ 12 เป็นต้นไป การบรรยายไวยากรณ์สม่ำเสมอจะคงที่ ไม่มีการเปลี่ยนแปลง รวมทั้งสามารถยอมรับตัวอย่างได้ทั้งหมด ดังนั้นภาษาดังกล่าวจึงลู่เข้าที่ความยาวใดๆ ดังนั้นการบรรยายสม่ำเสมอที่ใช้สำหรับอธิบายข้อมูลสายลำดับของกฎแฝง คือ

$$r_2r_2{}^*r_1{}^*r_6+r_2r_2{}^*r_1{}^*r_5r_7r_{10}$$ □

จากตัวอย่างที่ 3.4 การบรรยายสม่ำเสมอ $r_2r_2{}^*r_1{}^*r_6+r_2r_2{}^*r_1{}^*r_5r_7r_{10}$ ที่ได้จากอัลกอริทึมสำหรับสร้างไวยากรณ์อธิบายข้อมูลสายลำดับของกฎแฝง สามารถใช้อธิบายภาษา $a^m b^n$ เมื่อ $m \geq n \geq 1$ ซึ่งเป็นภาษาไม่พึ่งบริบท โดยข้อมูลตัวอย่างบวกทั้งหมดที่เป็นสมาชิกในภาษาดังกล่าวจะถูกยอมรับจากการบรรยายสม่ำเสมอที่สร้างขึ้น ส่วนข้อมูลตัวอย่างลบทั้งหมดที่ไม่ได้เป็นสมาชิกในภาษาดังกล่าว จะถูกปฏิเสธ

ตัวอย่างที่ 3.5 *ให้เซตของสายอักขระของตัวอย่างที่ต้องการเรียนรู้เป็นดังนี้ {ε , aca, bcb, aacaa, abcba, bacab, bbcbb, aaacaaa, aabcbaa, abacaba, abbcbba, baacaab, babcbab, bbacabb, bbbcbbb}*

วิธีทำ

กำหนดให้ $G = (V, \Sigma, P, S)$ จำนวนตัวแปร $|V| = 2$ และ $l_{\max} = 3$

$V = \{S_0, S_1\}, S = S_0$

P เป็นเซตจำกัดของกฎไวยากรณ์ไม่พึ่งบริบท เป็นดังนี้

$S_0 \rightarrow aS_0a \mid aS_0b \mid aS_0c \mid bS_0a \mid bS_0b \mid bS_0c \mid$

$r_1 \quad r_2 \quad r_3 \quad r_4 \quad r_5 \quad r_6$

$cS_0a \mid cS_0b \mid cS_0c \mid S_1 \mid \varepsilon$

$r_7 \quad r_8 \quad r_9 \quad r_{10} \quad r_{11}$

$S_1 \rightarrow aS_1 \mid bS_1 \mid cS_1 \mid S_0 \mid \varepsilon$

$r_{12} \quad r_{13} \quad r_{14} \quad r_{15} \quad r_{16}$ *(เมื่อ* $r_i \in P$ *)*

ตารางที่ 3.2 การแจงส่วนแบบเอียร์เลย์ข้อมูลตัวอย่างบวกตามลำดับในตัวอย่างที่ 3.5

ลำดับที่	สายอักขระตัวอย่างบวก	สายลำดับของกฎแฝง
1	ε	r_{11}
2	c	$r_{10}r_{14}r_{16}$
3	aca	$r_1r_{10}r_{14}r_{16}$
4	bcb	$r_5r_{10}r_{14}r_{16}$
5	aacaa	$r_1r_1r_{10}r_{14}r_{16}$
6	abcba	$r_1r_5r_{10}r_{14}r_{16}$
7	bacab	$r_5r_1r_{10}r_{14}r_{16}$
8	bbcbb	$r_5r_5r_{10}r_{14}r_{16}$
9	aaacaaa	$r_1r_1r_1r_{10}r_{14}r_{16}$
10	aabcbaa	$r_1r_1r_5r_{10}r_{14}r_{16}$
11	abacaba	$r_1r_5r_1r_{10}r_{14}r_{16}$
12	abbcbba	$r_1r_5r_5r_{10}r_{14}r_{16}$
13	baacaab	$r_5r_1r_1r_{10}r_{14}r_{16}$
14	babcbab	$r_5r_1r_5r_{10}r_{14}r_{16}$
15	bbacabb	$r_5r_5r_1r_{10}r_{14}r_{16}$
16	bbbcbbb	$r_5r_5r_5r_{10}r_{14}r_{16}$

รอบที่ 1 : จากข้อมูลสายลำดับของกฎแฝง r_{11} ที่ได้จากตัวอย่างบวกตัวแรก คือ ε ทำการผสานการบรรยายไวยากรณ์สม่ำเสมอเป็น

$$r_{11}$$

รอบที่ 2 : จากข้อมูลสายลำดับของกฎแฝง $r_{10}r_{14}r_{16}$ ที่ได้จากตัวอย่างบวกตัวที่ 2 คือ c ทำการผสานการบรรยายไวยากรณ์สม่ำเสมอเป็น

$$r_{11}+r_{10}r_{14}r_{16}$$

รอบที่ 3 : จากข้อมูลสายลำดับของกฎแฝง $r_1r_{10}r_{14}r_{16}$ ที่ได้จากตัวอย่างบวกตัวที่ 3 คือ *aca* ทำการผสานและลดรูปการบรรยายไวยากรณ์สม่ำเสมอเป็น

$$r_{11}+(\varepsilon+r_1)(r_{10}r_{14}r_{16})$$

รอบที่ 4 : จากข้อมูลสายลำดับของกฎแฝง $r_5r_{10}r_{14}r_{16}$ ที่ได้จากตัวอย่างบวกตัวที่ 4 คือ *bcb* ทำการผสานและลดรูปการบรรยายไวยากรณ์สม่ำเสมอเป็น

$$r_{11}+(\varepsilon+r_1+r_5)(r_{10}r_{14}r_{16})$$

รอบที่ 5 : จากข้อมูลสายลำดับของกฎแฝง $r_1r_1r_{10}r_{14}r_{16}$ ที่ได้จากตัวอย่างบวกตัวที่ 5 คือ *aacaa* ทำการผสานและลดรูปการบรรยายไวยากรณ์สม่ำเสมอเป็น

$$r_{11}+(\varepsilon+r_1+r_5+r_1r_1)(r_{10}r_{14}r_{16})$$

รอบที่ 6 : จากข้อมูลสายลำดับของกฎแฝง $r_1r_5r_{10}r_{14}r_{16}$ ที่ได้จากตัวอย่างบวกตัวที่ 6 คือ *abcba* ทำการผสานและลดรูปการบรรยายไวยากรณ์สม่ำเสมอเป็น

$$r_{11}+(\varepsilon+r_1+r_5+r_1r_1+r_1r_5)(r_{10}r_{14}r_{16})$$

รอบที่ 7 : จากข้อมูลสายลำดับของกฎแฝง $r_5r_1r_{10}r_{14}r_{16}$ ที่ได้จากตัวอย่างบวกตัวที่ 7 คือ *bacab* ทำการผสานและลดรูปการบรรยายไวยากรณ์สม่ำเสมอเป็น

$$r_{11}+(\varepsilon+r_1+r_5+r_1r_1+r_1r_5+r_5r_1)(r_{10}r_{14}r_{16})$$

รอบที่ 8 : จากข้อมูลสายลำดับของกฎแฝง $r_5r_5r_{10}r_{14}r_{16}$ ที่ได้จากตัวอย่างบวกตัวที่ 8 คือ *bbcbb* ทำการผสานและลดรูปการบรรยายไวยากรณ์สม่ำเสมอเป็น

$$r_{11}+(r_1+r_5)^*(r_{10}r_{14}r_{16})$$

รอบที่ 9 : จากข้อมูลสายลำดับของกฎแฝง $r_1r_1r_{10}r_{14}r_{16}$ ที่ได้จากตัวอย่างบวกตัวที่ 9 คือ *aaacaaa* การบรรยายไวยากรณ์สม่ำเสมอไม่เปลี่ยนแปลง

รอบที่ 10 : จากข้อมูลสายลำดับของกฎแฝง $r_1r_1r_5r_{10}r_{14}r_{16}$ ที่ได้จากตัวอย่างบวกตัวที่ 10 คือ *aabcbaa* การบรรยายไวยากรณ์สม่ำเสมอไม่เปลี่ยนแปลง

รอบที่ 11 : จากข้อมูลสายลำดับของการใช้กฎ $r_1r_5r_1r_{10}r_{14}r_{16}$ ที่ได้จากตัวอย่างบวกตัวที่ 11 คือ *abacaba* การบรรยายไวยากรณ์สม่ำเสมอไม่เปลี่ยนแปลง

รอบที่ 12 : จากข้อมูลสายลำดับของกฎแฝง $r_1r_5r_5r_{10}r_{14}r_{16}$ ที่ได้จากตัวอย่างบวกตัวที่ 12 คือ *abbcbba* การบรรยายไวยากรณ์สม่ำเสมอไม่เปลี่ยนแปลง

รอบที่ 13 : จากข้อมูลสายลำดับของกฎแฝง $r_5r_1r_1r_{10}r_{14}r_{16}$ ที่ได้จากตัวอย่างบวกตัวที่ 13 คือ *baacaab* การบรรยายไวยากรณ์สม่ำเสมอไม่เปลี่ยนแปลง

รอบที่ 14 : จากข้อมูลสายลำดับของกฎแฝง $r_5r_1r_5r_{10}r_{14}r_{16}$ ที่ได้จากตัวอย่างบวกตัวที่ 14 คือ *babcbab* การบรรยายไวยากรณ์สม่ำเสมอไม่เปลี่ยนแปลง

รอบที่ 15 : จากข้อมูลสายลำดับของกฎแฝง $r_5r_5r_1r_{10}r_{14}r_{16}$ ที่ได้จากตัวอย่างบวกตัวที่ 15 คือ *bbacabb* การบรรยายไวยากรณ์สม่ำเสมอไม่เปลี่ยนแปลง

รอบที่ 16 : จากข้อมูลสายลำดับของกฎแฝง $r_5r_5r_5r_{10}r_{14}r_{16}$ ที่ได้จากตัวอย่างบวกตัวที่ 16 คือ *bbbcbbb* การบรรยายไวยากรณ์สม่ำเสมอไม่เปลี่ยนแปลง

สังเกตเห็นว่าตั้งแต่สายลำดับของกฎแฝงตั้งแต่รอบที่ 9 เป็นต้นไป การบรรยายสม่ำเสมอจะคงที่ ไม่มีการเปลี่ยนแปลง รวมทั้งสามารถยอมรับตัวอย่างได้ทั้งหมด ดังนั้นภาษาดังกล่าวจึงลู่เข้าที่ความยาวใดๆ ดังนั้นการบรรยายสม่ำเสมอที่ใช้สำหรับอธิบายข้อมูลสายลำดับของกฎแฝง คือ

$$r_{11}+(r_1+r_5)^*(r_{10}r_{14}r_{16}) \qquad \square$$

จากตัวอย่างที่ 3.5 การบรรยายสม่ำเสมอ $r_{11}+(r_1+r_5)^*(r_{10}r_{14}r_{16})$ ที่ได้จากอัลกอริทึมสำหรับสร้างไวยากรณ์อธิบายข้อมูลสายลำดับของกฎแฝง สามารถใช้อธิบายภาษาพาลินโดรม ซึ่งเป็นภาษาไม่พึ่งบริบท โดยข้อมูลตัวอย่างบวกทั้งหมดที่เป็นสมาชิกในภาษาพาลินโดรมจะถูกยอมรับจากการบรรยายสม่ำเสมอที่สร้างขึ้น ส่วนข้อมูลตัวอย่างลบที่ไม่ได้เป็นสมาชิกในภาษาพาลินโดรมทั้งหมดจะถูกปฏิเสธ

3.4 ผลการวิเคราะห์

เนื่องจากมีทฤษฎีเกี่ยวกับภาษาสม่ำเสมอซึ่งพิสูจน์ไว้แล้วว่า ภาษาสม่ำเสมอจะมีคุณสมบัติดังทฤษฎีที่ 2.3 และจากการวิเคราะห์อัลกอริทึมและคุณสมบัติของภาษาสม่ำเสมอเราได้เสนอนิยามและผลวิเคราะห์เป็นทฤษฎีดังนี้

นิยามที่ 3.3 *S_P เรียกว่าเป็น เซตตัวอย่างบวกที่มีโครงสร้างสมบูรณ์ ของภาษาสม่ำเสมอ L โดยที่ L ยอมรับได้โดยออโตมาตาที่มีจำนวนสถานะเท่ากับ n ก็ต่อเมื่อ ทุก x ที่เป็นสมาชิกของ L ถ้า $|x| \leq 2n$ แล้ว x ต้องเป็นสมาชิก S_P ด้วย*

ทฤษฎีบทที่ 3.1 *ถ้าภาษา L เป็นภาษาสม่ำเสมอ และ S_P เป็นตัวอย่างบวกที่มีโครงสร้างสมบูรณ์แล้ว อัลกอริทึมสามารถสร้างไวยากรณ์ไม่พึ่งบริบทโดยใช้ลำดับของกฎแฝงได้ทุกภาษา L*

พิสูจน์

สมมติให้ L เป็นภาษาสม่ำเสมอ จะต้องมีออโตมาตาที่มีจำนวนสถานะ n ดังนั้นย่อมมีสายอักขระที่ยอมรับออโตมาตาที่มีความยาวไม่เกิน n โดยที่ไม่เกิดการวนซ้ำขึ้น ดังนั้นสายอักขระที่ยาวกว่า n ต้องเกิดการวนซ้ำ และการวนซ้ำหนึ่งครั้งนั้นจะมีความยาวไม่เกิน n ทำให้เกิดสายอักขระที่มีความยาวไม่เกิน 2n

เนื่องจากหลักการทำงานของอัลกอริทึมสำหรับสร้างภาษาสำหรับอธิบายข้อมูลสายลำดับของกฎแฝง จะทำการพิจารณาตัวอย่างบวกที่มีโครงสร้างสมบูรณ์ที่ถูกแจงส่วนแบบเอียร์เลย์ที่เข้ามา ดังนั้นจากทฤษฎีบทที่ 2.3 ถ้าเราทดสอบการวนซ้ำของการใช้กฎไวยากรณ์เริ่มต้นจะได้ว่า เราจะสามารถค้นพบรูปแบบการวนซ้ำของไวยากรณ์สม่ำเสมอที่สามารถสร้างสายอักขระใดๆ ที่มีความยาวมากกว่า 2n และเป็นสมาชิกของ L ได้เสมอ ■

ทฤษฎีบทที่ 3.2 *อัลกอริทึมในการสร้างไวยากรณ์ไม่พึ่งบริบทโดยใช้ลำดับของกฎแฝงสามารถทำงานโดยใช้ความซับซ้อนเชิงเวลาไม่เกินฟังก์ชันพหุนาม*

พิสูจน์

เนื่องจากอัลกอริทึมสร้างไวยากรณ์ไม่พึ่งบริบทแบบเชื่อมตรงโดยใช้ลำดับของกฎแฝงจากข้อมูลตัวอย่าง การทำงานของอัลกอริทึมโดยรวมแบ่งเป็นอัลกอริทึมย่อยๆ ดังนั้นเราจึงต้องแยกการพิจารณาออกเป็นส่วนๆ แล้วจึงมาหาความซับซ้อนเชิงเวลารวม

การพิจารณาความซับซ้อนเชิงเวลาในอัลกอริทึมสำหรับสร้างไวยากรณ์ไม่พึ่งบริบทเริ่มต้น ที่จะเลือกไวยากรณ์ไม่พึ่งบริบทเริ่มต้นแบบที่ 1 จะเห็นว่าใช้ความซับซ้อนเชิงเวลาเท่ากับ $O(|P|)$ เมื่อ $|P|$ เป็นจำนวนกฎ ซึ่งมีค่าคงที่ เนื่องจากการทำงานของอัลกอริทึมเป็นการทำงานแบบรอบเดียว

การพิจารณาความซับซ้อนเชิงเวลาในอัลกอริทึมสำหรับดึงลำดับการใช้กฎแฝงจากการแจงส่วนแบบเอียร์เลย์ ใช้ข้อมูลตัวอย่างทั้งหมดในการเรียนรู้ ซึ่งใช้ความซับซ้อนเชิงเวลาเท่ากับ $O(\| S_P \|^3)$ เมื่อ $\| S_P \|$ เป็นผลรวมของความยาวสายอักขระทุกตัวที่เป็นสมาชิกของข้อมูลตัวอย่าง

การพิจารณาความซับซ้อนเชิงเวลาในอัลกอริทึมสำหรับสร้างไวยากรณ์อธิบายข้อมูลสายลำดับของกฎแฝง ใช้ความซับซ้อนเชิงเวลาเท่ากับ $O(\| S_{PT} \|^3)$ ซึ่งจะคล้ายกับเวลาที่ใช้ในอัลกอริทึมอาร์พีเอ็นไอที่ใช้สร้างออโตมาตาจำกัดสถานะเชิงกำหนดในกรณีที่พิจารณาเพียงตัวอย่างบวกอย่างเดียว เมื่อ $\| S_{PT} \|$ เป็นผลรวมของความยาวสายลำดับของกฎแฝงทุกตัวที่เป็นสมาชิกของข้อมูลตัวอย่างที่ผ่านการแจงส่วนแบบเอียร์เลย์

ดังนั้นเมื่อพิจารณาความซับซ้อนรวมทุกขั้นตอนของอัลกอริทึมจะได้ความซับซ้อนเชิงเวลาเท่ากับ $O(| P |) + O(\| S_P \|^3) + O(\| S_{PT} \|^3)$ และการแจงส่วนแบบเอียร์เลย์พบว่า ถ้ากฎไวยากรณ์ $A \to \alpha$ *เมื่อ* $\alpha \in \Sigma^* V \Sigma^* \cup \varepsilon$ แล้ว $| S_{PT} | \leq | S_P |$ เสมอ ดังนั้นความซับซ้อนเชิงเวลารวมจึงเท่ากับ $O(\| S_P \|^3)$ ซึ่งอยู่ในฟังก์ชันพหุนาม ■

3.5 สรุปบท

จากเนื้อหาทั้งหมดที่ได้กล่าวถึงในบทนี้ การวิเคราะห์รูปแบบของปัญหา และการแบ่งการทำงานของอัลกอริทึมออกเป็นอัลกอริทึมย่อยๆ ได้แก่ อัลกอริทึมสำหรับสร้างไวยากรณ์ไม่พึ่งบริบทเริ่มต้น อัลกอริทึมสำหรับดึงลำดับการใช้กฎแฝงจากการแจงส่วนแบบเอียร์เลย์ และอัลกอริทึมสำหรับสร้างไวยากรณ์อธิบายข้อมูลสายลำดับของกฎแฝง ทำให้สามารถนำไปใช้งานกับข้อมูลตัวอย่างได้จริง รวมทั้งจากผลการวิเคราะห์ อัลกอริทึมนี้ใช้ความซับซ้อนเชิงเวลาระดับพหุนาม และสามารถนำไปใช้ในการสร้างไวยากรณ์สำหรับอธิบายทุกภาษาสม่ำเสมอและบางภาษาไม่พึ่งบริบทได้อย่างสมเหตุสมผล

บทที่ 4

ผลการทดลอง

เนื่องจากงานวิจัยนี้ได้นำเสนอไวยากรณ์ไม่พึ่งบริบทที่ใช้การเรียนรู้รูปแบบลำดับของกฎแฝงจากข้อมูลตัวอย่างบวกเพียงอย่างเดียว เพื่อแยกแยะข้อมูลตัวอย่างบวกและข้อมูลตัวอย่างลบได้อย่างถูกต้อง โดยทำการทดสอบกับภาษาตัวอย่าง แล้วสร้างเซตของข้อมูลตัวอย่างบวกที่มีโครงสร้างสมบูรณ์ ซึ่งในงานวิจัยของวุฒิ สุนทรภัณฑ์ และงานวิจัยของนากามุระและมัตซุโมโต ได้ใช้ตัวอย่างภาษาเหล่านี้ในการทดลอง ดังตารางที่ 4.1 ซึ่งในตารางได้อธิบายภาษา ตัวอย่างภาษา และคุณสมบัติของภาษา ซึ่งภาษาที่ 1, 2 เป็นภาษาสม่ำเสมอ ส่วนภาษาที่ 3-9 เป็นภาษาไม่พึ่งบริบท

ตารางที่ 4.1 ตัวอย่างภาษาที่ใช้ในการทดสอบ

ภาษาที่	ภาษา	ตัวอย่างภาษา	คุณสมบัติ
1	(aa)*	aa, aaaa, aaaaaa	regular
2	(ab)*	ab, abab, ababab	regular
3	$a^m b^n$ $(m \geq n \geq 1)$	ab, aab, aaab, aaabb	context-free
4	balanced parentheses	(), (()), ()(), (())()	context-free
5	$\{w = w^R \mid w \in \{a,b\}^+\}$	a, b, aa, bb, aaa, aba, bab	context-free
6	palindrome with center mark	aca, bcb, aacaa, abcba	context-free
7	number of a's = number of b's	ab, ba, aabb, abab	context-free
8	number of a's = 2 x number of b's	aab, aba, baa, aaaabb	context-free
9	$\{a^i b^j c^k \mid i = j \text{ or } j = k, i,j,k > 0\}$	abc, aabc, abcc, aabbc	context-free

ในการเรียนรู้ตัวอย่างภาษา จะใช้ข้อมูลตัวอย่างที่มีความยาวไม่เกิน 30 ตัวอักษร

จากอัลกอริทึมสำหรับสร้างไวยากรณ์ไม่พึ่งบริบทเริ่มต้น โดยพิจารณาจำนวนอักขระทั้งหมดที่ต้องใช้ในแต่ละภาษา สามารถสร้างไวยากรณ์เริ่มต้นได้ ดังนี้

สำหรับภาษาที่มีเซตอักขระ $\Sigma = \{a,b\}$ ได้แก่ 1, 2, 3, 5, 7 และ 8 โดยกำหนดไวยากรณ์เริ่มต้นที่เลือกใช้ เป็นดังนี้

กำหนดให้ $G_1 = (V_1, \Sigma, P_1, S)$

จำนวนตัวแปร $|V_1| = 2$ และ $l_{\max} = 3$

$V_1 = \{S_0, S_1\}, S = S_0$

P_1 เป็นเซตจำกัดของกฎไวยากรณ์ไม่พึ่งบริบท เป็นดังนี้

$S_0 \rightarrow aS_0a \mid aS_0b \mid bS_0a \mid bS_0b \mid S_1 \mid \varepsilon$

$r_1 \quad r_2 \quad r_3 \quad r_4 \quad r_5 \quad r_6$

$S_1 \rightarrow aS_1 \mid bS_1 \mid S_0 \mid \varepsilon$

$r_7 \quad r_8 \quad r_9 \quad r_{10}$ *(เมื่อ* $r_i \in P_1$*)*

สำหรับภาษาที่มีเซตอักขระ $\Sigma = \{a,b,c\}$ ได้แก่ ภาษาที่ 6 และ 9 โดยกำหนดไวยากรณ์เริ่มต้นที่เลือกใช้ เป็นดังนี้

กำหนดให้ $G_2 = (V_2, \Sigma_2, P_2, S), |V_2| = 2$ และ $l_{\max} = 3$

$V_2 = \{S_0, S_1\}, S = S_0$

P_2 เป็นเซตจำกัดของกฎไวยากรณ์ไม่พึ่งบริบท ดังนี้

$S_0 \rightarrow aS_0a \mid aS_0b \mid aS_0c \mid bS_0a \mid bS_0b \mid bS_0c \mid$

$r_1 \quad r_2 \quad r_3 \quad r_4 \quad r_5 \quad r_6$

$cS_0a \mid cS_0b \mid cS_0c \mid S_1 \mid \varepsilon$

$r_7 \quad r_8 \quad r_9 \quad r_{10} \quad r_{11}$

$S_1 \rightarrow aS_1 \mid bS_1 \mid cS_1 \mid S_0 \mid \varepsilon$

$r_{12} \quad r_{13} \quad r_{14} \quad r_{15} \quad r_{16}$ *(เมื่อ* $r_i \in P_2$*)*

สำหรับภาษาที่มีเซตอักขระ $\Sigma = \{(,)\}$ ได้แก่ ภาษาที่ 4 โดยกำหนดไวยากรณ์เริ่มต้นที่เลือกใช้ เป็นดังนี้

กำหนดให้ $G_3 = (V_3, \Sigma_1, P_3, S), |V_3| = 2$ และ $l_{\max} = 3$

$V_3 = \{S_0, S_1\}, S = S_0$ และ

P_3 เป็นเซตจำกัดของกฎไวยากรณ์ไม่พึ่งบริบท ดังนี้

$S_0 \rightarrow (S_0(\mid (S_0) \mid)S_0(\mid)S_0) \mid S_1 \mid \varepsilon$

$r_1 \quad r_2 \quad r_3 \quad r_4 \quad r_5 \quad r_6$

$S_1 \rightarrow (S_1 \mid)S_1 \mid S_0 \mid \varepsilon$

$r_7 \quad r_8 \quad r_9 \quad r_{10}$ *(เมื่อ* $r_i \in P_3$*)*

ตารางที่ 4.2 ผลการทดสอบอัลกอริทึมกับตัวอย่างภาษา

ภาษาที่	ภาษา	ภาษาสำหรับอธิบายลำดับของกฎแฝง	ไวยากรณ์เริ่มต้นที่ใช้	ความยาวตัวอย่างที่ใช้เพื่อลู่เข้า	จำนวนตัวอย่างที่ใช้เพื่อลู่เข้า
1	(aa)*	$r_1{}^*r_6$	แบบที่ 1	4	3
2	(ab)*	$(r_2r_3)^*r_6+(r_2r_3)^*r_2r_6$	แบบที่ 1	10	6
3	a^mb^n $(m \geq n \geq 1)$	$r_2(r_2{}^*r_1{}^*r_5r_7r_{10}+r_2{}^*r_1{}^*r_6)$	แบบที่ 1	7	11
4	balanced parentheses	-	แบบที่ 3	-	นิยามได้จำกัดระดับความยาว
5	$\{w = w^R \mid w \in \{a,b\}^+\}$	$(r_1+r_4)^*r_5(r_7+r_8)r_{10}+(r_1+r_4)^*r_6$	แบบที่ 1	5	19
6	palindrome with center mark	$r_{11}+(r_1+r_5)^*r_{10}r_{14}r_{16}$	แบบที่ 2	5	8
7	number of a's = number of b's	-	แบบที่ 1	-	นิยามได้จำกัดระดับความยาว
8	number of a's = 2 x number of b's	-	แบบที่ 1	-	นิยามได้จำกัดระดับความยาว
9	$\{a^ib^jc^k \mid i = j \text{ or } j = k, i,j,k > 0\}$	-	แบบที่ 2	-	นิยามได้จำกัดระดับความยาว

จากตารางที่ 4.2 แสดงให้เห็นผลการทดสอบอัลกอริทึมกับภาษาต่างๆ ประกอบไปด้วย ภาษาสำหรับอธิบายลำดับของกฎแฝง ไวยากรณ์เริ่มต้นที่ใช้ ความยาวตัวอย่างที่ใช้เพื่อลู่เข้า และจำนวนตัวอย่างที่ใช้เพื่อลู่เข้า จะเห็นว่าจากการทำงานของอัลกอริทึม ภาษาที่สามารถสร้างไวยากรณ์ให้ลู่เข้าคือ ภาษาที่ 1, 2, 3, 5, 6 ซึ่งไวยากรณ์ที่ได้นำไปใช้แยกแยะข้อมูลตัวอย่างบวกและตัวอย่างลบได้อย่างถูกต้อง ส่วนภาษาที่ไม่สามารถสร้างไวยากรณ์ให้ลู่เข้าคือ ภาษาที่ 4, 7, 8, 9 ซึ่งไวยากรณ์ที่ได้นำไปใช้แยกแยะข้อมูลตัวอย่างบวกและตัวอย่างลบที่มีความยาวไม่เกินไปกว่าความยาวสูงสุดของข้อมูลตัวอย่างบวกที่ใช้ในการเรียนรู้

พิจารณาจากกลุ่มภาษาที่ 1, 2 นั้นเป็นภาษาสม่ำเสมอ และจากทฤษฎีบทที่ 3.1 ทำให้กลุ่มภาษาดังกล่าวสามารถสร้างไวยากรณ์ที่ลู่เข้าได้

ส่วนกลุ่มภาษาที่ 3, 5, 6 จากการวิเคราะห์พบว่า กลุ่มภาษาดังกล่าวสามารถเขียนกฎไวยากรณ์ไม่พึ่งบริบทให้อยู่ในรูปแบบตามนิยามที่ 2.19, 2.20 และ 2.21 ได้ ซึ่งสอดคล้องกับอัลกอริทึมสำหรับสร้างไวยากรณ์ไม่พึ่งบริบทเริ่มต้น ทำให้กลุ่มภาษาดังกล่าวสามารถสร้างไวยากรณ์ที่ลู่เข้าได้

แต่ในทางกลับกัน ในกลุ่มภาษาที่ 4, 7, 8, 9 กลุ่มภาษาดังกล่าว ไม่สามารถเขียนกฎไวยากรณ์ไม่พึ่งบริบทให้อยู่ในรูปแบบตามนิยามที่ 2.19, 2.20 และ 2.21 ได้ ซึ่งไม่สอดคล้องกับอัลกอริทึมสำหรับสร้างไวยากรณ์ไม่พึ่งบริบทเริ่มต้น ที่กำหนดรูปแบบของกฎทางฝั่งขวาเป็น $\alpha \in \Sigma^* V \Sigma^* \cup \varepsilon$ ทำให้กลุ่มภาษาดังกล่าวไม่สามารถสร้างไวยากรณ์ที่ลู่เข้าได้

ทำการเปรียบเทียบผลการทดลองกับงานวิจัยของวุฒิ สุนทรภัณฑ์ [4] ที่ใช้การเรียนรู้โดยใช้การแทนสายอักขระและผสานด้วยตัวแปร และงานวิจัยของนากามุระและมัตซุโมโต [3] ที่ใช้การเรียนรู้ด้วยวิธีซีวายเค ได้ดังตารางที่ 4.3 จะเห็นว่าในการเรียนรู้บางภาษาจำนวนตัวอย่างที่ใช้เพื่อลู่เข้าของอัลกอริทึมของเรานั้นใช้จำนวนตัวอย่างที่น้อยกว่า

ตารางที่ 4.3 แสดงผลการเปรียบเทียบกับงานวิจัยของวุฒิ และงานวิจัยของนากามุระและมัตซุโมโต

ภาษาที่	ภาษา	จำนวนตัวอย่างที่ใช้ในการลู่เข้าจากการเรียนรู้		
		เรียนรู้โดยใช้ลำดับของกฎแฝง	งายวิจัยวุฒิ [4]	งานวิจัยของนากามุระและมัตซุโมโต [3]
1	(aa)*	3	30	30
2	(ab)*	6	30	30
3	$a^m b^n \ (m \geq n \geq 1)$	11	30	30
4	balanced parentheses	-	30	30
5	$\{w = w^R \mid w \in \{a,b\}^+\}$	19	14	14
6	palindrome with center mark	8	14	14
7	number of a's = number of b's	-	30	30
8	number of a's = 2 x number of b's	-	-	126
9	$\{a^i b^j c^k \mid i = j \text{ or } j = k, i,j,k > 0\}$	-	62	-

บทที่ 5

สรุปและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ในงานวิจัยนี้อัลกอริทึมสำหรับสร้างไวยากรณ์ไม่พึ่งบริบทแบบเชื่อมตรงโดยใช้ลำดับของกฎแฝงจากข้อมูล โดยข้อมูลตัวอย่างบวกที่ใช้เรียนรู้ต้องมีโครงสร้างสมบูรณ์ โดยกำหนดความสำเร็จของการเรียนรู้ด้วยหลักการจำแนกภาษาภายในจำกัด ซึ่งข้อกำหนดสำคัญคือ ไวยากรณ์ที่ได้สามารถตอบความเป็นสมาชิกในภาษาจากตัวอย่างบวกได้ถูกต้องทั้งหมด รวมทั้งเมื่อตัวอย่างลบที่ไม่อยู่ในภาษานำมาใช้ทดสอบต้องสามารถตอบความไม่เป็นสมาชิกในภาษาได้

ความสามารถของอัลกอริทึม คือ สามารถเรียนรู้ไวยากรณ์ให้ลู่เข้าด้วยหลักการจำแนกภาษาภายในจำกัด โดยกลุ่มภาษาที่เรียนรู้แล้วสำเร็จ ได้แก่ ทุกภาษาสม่ำเสมอ และกลุ่มภาษาไม่พึ่งบริบทที่เซตของกฎไวยากรณ์ $A \to \alpha$ สามารถเขียนให้อยู่ในรูป $\alpha \in \Sigma^* V \Sigma^* \cup \Sigma^*$ ได้ ซึ่งกลุ่มภาษาไม่พึ่งบริบทกลุ่มดังกล่าว ได้แก่ กลุ่มภาษาไม่พึ่งบริบทเชิงเส้น กลุ่มภาษาไม่พึ่งบริบทเชิงเส้นแบบคู่ และกลุ่มภาษาเชิงเส้นเชิงกำหนด ส่วนข้อจำกัดของอัลกอริทึมคืออัลกอริทึมไม่สามารถสร้างไวยากรณ์ที่ลู่เข้าได้ในบางภาษาไม่พึ่งบริบท เมื่อเรียนรู้ข้อมูลตัวอย่างไปช่วงระยะเวลาหนึ่งจะพบว่า การบรรยายด้วยไวยากรณ์สม่ำเสมอสายลำดับกฎแฝงจะไม่คงที่ มีการเปลี่ยนแปลงอยู่ตลอดเวลา ซึ่งกลุ่มภาษาที่เรียนรู้แล้วไม่ลู่เข้า ได้แก่ กลุ่มภาษาไม่พึ่งบริบทที่เซตของกฎ $A \to \alpha$ ไม่สามารถเขียนให้อยู่ในรูป $\alpha \in \Sigma^* V \Sigma^* \cup \Sigma^*$

จากทฤษฎีที่ 3.2 อัลกอริทึมการสร้างไวยากรณ์ไม่พึ่งบริบทโดยใช้ลำดับของกฎแฝงมีประสิทธิภาพสามารถจำแนกภาษาภายในจำกัดเชิงเวลาพหุนาม ใช้ความซับซ้อนเชิงเวลาเท่ากับ $O(\| S_P \|^3)$ เมื่อ $\| S_P \|$ เป็นผลรวมของความยาวสายอักขระทุกตัวที่เป็นสมาชิกของเซตตัวอย่างบวก เมื่อเทียบกับอัลกอริทึมของวุฒิ สุนทรภัณฑ์ [4] ใช้ความซับซ้อนเชิงเวลาเท่ากับ $O(\| S_P \|^2 \| S_N \|^3)$ เมื่อ $\| S_N \|$ เป็นผลรวมของความยาวสายอักขระทุกตัวที่เป็นสมาชิกของเซตตัวอย่างลบ และอัลกอริทึมของนากามุระและมัตซุโมโต [3] ที่ใช้ความซับซ้อนเชิงเวลาระดับเลขชี้กำลัง ดังนั้นจึงสรุปได้ว่า อัลกอริทึมที่ได้ต้องมีความซับซ้อนเชิงเวลาที่น้อยกว่า เมื่อเทียบกับอัลกอริทึมของวุฒิ สุนทรภัณฑ์ [4] และอัลกอริทึมของนากามุระและมัตซุโมโต [3]

การทำงานของอัลกอริทึมนี้จะไม่รองรับข้อมูลตัวอย่างที่ขาดความสมบูรณ์ หรือข้อมูลตัวอย่างที่มีค่าผิดพลาด เนื่องจากในงานวิจัยนี้สนใจการเรียนรู้ไวยากรณ์ด้วยลำดับของกฎแฝงจากข้อมูลตัวอย่างที่มีความสมบูรณ์เท่านั้น ดังนั้นหากต้องการนำอัลกอริทึมไปใช้กับข้อมูลที่ขาดความสมบูรณ์ จะต้องมีการปรับปรุงการทำงานของอัลกอริทึมให้รองรับค่าความผิดพลาดที่เกิดขึ้น ด้วยหลักการทางสถิติ

5.2 ข้อเสนอแนะ

จากงานวิจัยนี้เป็นงานวิจัยเสนอการสร้างไวยากรณ์ไม่พึ่งบริบทแบบใหม่โดยใช้ลำดับของกฎแฝงซึ่งพบว่า ยังสามารถทำการวิจัยเพื่อปรับปรุงและพัฒนาให้งานวิจัยมีประสิทธิภาพการทำงานมากยิ่งขึ้น ซึ่งแยกข้อเสนอแนะเป็นข้อๆ ดังนี้

5.2.1 จากงานวิจัยนี้ทำให้เกิดแนวคิดใหม่ในส่วนของอัลกอริทึมสำหรับสร้างไวยากรณ์ไม่พึ่งบริบทเริ่มต้น โดยที่กฎไวยากรณ์เริ่มต้นนั้นสามารถปรับเปลี่ยนแปลงได้ เมื่อเรียนรู้ข้อมูลตัวอย่างใหม่ที่รับเข้ามา รวมทั้งแนวคิดในการเสนอแนวทางการสร้างไวยากรณ์ไม่พึ่งบริบทเริ่มต้นให้รองรับทุกภาษาไม่พึ่งบริบท

5.2.2 การนำอัลกอริทึมนี้ไปใช้กับข้อมูลจริง ซึ่งพบว่าข้อมูลจริงที่ใช้ในการเรียนรู้อาจมีความผิดพลาดเกิดขึ้นได้ ดังนั้นอัลกอริทึมที่ใช้ในการสร้างไวยากรณ์จึงควรปรับปรุงให้สามารถสร้างไวยากรณ์เพื่อรองรับข้อมูลที่ไม่มีความสมบูรณ์นี้ด้วย

รายการอ้างอิง

[1] Earley, J. “An efficient context-free parsing algorithm”. Communications of the ACM 13 (February 1970): 94-102.

[2] Gold, E.M. “Language identification in the limit”. Information and control 10 (1967): 447-474.

[3] Nakamura, K., and Matsumoto, M. “Incremental learning of context-free grammars based on bottom-up parsing and search”. Pattern recognition society 38 (2005): 1384-1392.

[4] วุฒิ สุทรภัณฑ์, “การปรับปรุงและพัฒนาอัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบท”, วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2549.

[5] Langley, P., and Stromsten, S. “Learning context-free grammars with a simplicity bias”. EMCL, LNCI 1810 (2000): 220-228.

[6] Hopcroft, J. E., Motwani R., and Ullman, J. D. Introduction to automata theory, languages, and computation (Addison-Wesley, 2001): 228-302.

[7] De La Higuera, C. “Characteristic sets for polynomial grammatical inference”. Machine learning journal 27 (1997): 125-138.

[8] Oncina J., and Garcia P. “Inferring regular languages in polynomial update time”. Pattern recognition and image analysis (1992): 49-61.

[9] Kearns, M., and Valiant L. “Cryptographic limitations on learning boolean formulae and finite automata” Journal of the ACM 21 (January 1994): 67-95.

[10] Sakakibara, Y. “Ga-based learning of context-free grammars using tabular representations”. ICML 16 (1999): 354-360.

[11] Sakakibara, Y. “Learning context-free grammars using tabular representations”. Pattern recognition 38 (2005): 1372-1383.

[12] Valiant, L. G. “A theory of the learnable”. Communications of the association for computing machinery 27 (1894): 1334-1142.

[13] de la Higuera, C. “Characteristic sets for polynomial grammatical inference”. Machine learning journal 27 (1997): 125-138.

ประวัติผู้เขียนวิทยานิพนธ์

นายสุรพงษ์ ผลประกอบศิลป์ เกิดเมื่อวันที่ 22 กันยายน พ.ศ. 2525 จบการศึกษาระดับมัธยมตอนปลายจากโรงเรียนสารสิทธิ์พิทยาลัย เข้าศึกษาต่อในระดับปริญญาบัณฑิต สาขาวิทยาการคอมพิวเตอร์และสารสนเทศ คณะวิทยาศาสตร์ประยุกต์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ จนสำเร็จการศึกษาในปี พ.ศ. 2546 และศึกษาต่อในระดับปริญญามหาบัณฑิต ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย