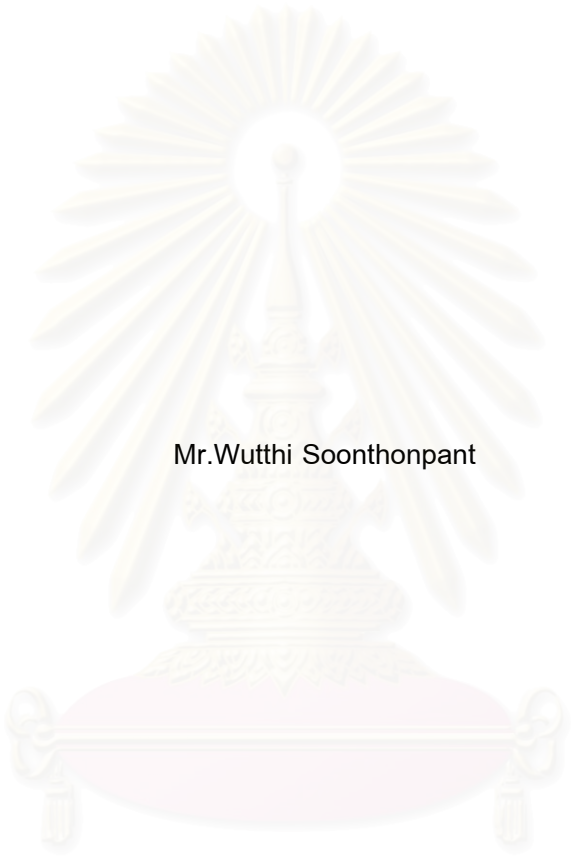
การปรับปรุงและพัฒนาอัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบท

นายวุฒิ สุนทรภัณฑ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2549

IMPROVING AND DEVELOPING CONTEXT FREE GRAMMAR INFERENCE ALGORITHM

Mr.Wutthi Soonthonpant



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 2006

หัวข้อวิทยานิพนธ์	การปรับปรุงและพัฒนาอัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบท
โดย	นายวุฒิ สุนทรภัณฑ์
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.สมชาย ประสิทธิ์จูดระกูล)

..... อาจารย์ที่ปรึกษา
(อาจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์)

..... กรรมการ
(รองศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ)

นายวุฒิ สุนทรภัณฑ์ : การปรับปรุงและพัฒนาอัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบท (IMPROVING AND DEVELOPING CONTEXT FREE GRAMMAR INFERENCE ALGORITHM) อาจารย์ที่ปรึกษา : อาจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์, 55 หน้า.

การอนุมานไวยากรณ์ไม่พึ่งบริบท เป็นส่วนสำคัญในการวิเคราะห์แก้ปัญหาการรู้จำรูปแบบ และงานวิจัยต่าง ๆที่เกี่ยวกับการประมวลผลภาษาธรรมชาติได้เน้นไปที่วิธีการพัฒนาการอนุมาน ซึ่งปัญหาหลักของการอนุมานไวยากรณ์ไม่พึ่งบริบท คือ ใช้ค่าความซับซ้อนเชิงเวลาของการอนุมานไวยากรณ์ที่สูง ซึ่งมีผลงานวิจัยทางทฤษฎีระบุว่าไม่สามารถหาอัลกอริทึมอนุมานไวยากรณ์ไม่พึ่งบริบทที่ใช้ความซับซ้อนเชิงเวลาไม่เกินฟังก์ชั่นพหุนามได้

ดังนั้นงานวิจัยนี้ได้นำเสนออัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบทสำหรับบางภาษาไม่พึ่งบริบทรวมทั้งภาษาสม่ำเสมอ ที่มีความซับซ้อนเชิงเวลาไม่เกินฟังก์ชั่นพหุนาม ซึ่งหลักการทำงานของอัลกอริทึมจะพิจารณาสร้างกฎวนซ้ำจากข้อมูลตัวอย่างที่อยู่ในภาษาเรียกว่าข้อมูลตัวอย่างบวก และเพื่อไม่ให้ไวยากรณ์ไม่พึ่งบริบทมีความกว้างมากเกินไป งานวิจัยนี้ได้นำข้อมูลตัวอย่างที่ไม่อยู่ในภาษาเรียกว่าตัวอย่างลบมาร่วมพิจารณาด้วย

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต วุฒิ สุนทรภัณฑ์
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษา Atharit Suraruks
ปีการศึกษา 2549

4670470421 : MAJOR COMPUTER SCIENCE
KEY WORD : GRAMMATICAL INFERENCE / CONTEXT-FREE GRAMMAR
WUTTHI SOONTHONPANT : IMPROVING AND DEVELOPING CONTEXT FREE GRAMMAR INFERENCE ALGORITHM. THESIS ADVISOR : ATHASIT SURARERKS, Ph.D., 55 pp.

Context-free grammar inference plays an important role in pattern recognition. Many researches in natural language processing are focused on how to improve an inference technique. In fact, the major problem is that it requires a high degree of computational complexity. Some theoretical results stated that the problem cannot be solved by any polynomial time algorithms.

This research is aimed to introduce an inference algorithm for some context-free languages including regular languages. The proposed algorithm needs a polynomial time complexity. Our concept is to create some recursive rules using positive samples. In order to avoid an overgeneralization problem, some negative samples are considered in the training process.

Department Computer Engineering Student's signature .....
Field of study Computer Science Advisor's signature .....
Academic year 2006

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความอนุเคราะห์ และความช่วยเหลืออย่างยิ่งจาก อาจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์ อาจารย์ที่ปรึกษา ซึ่งเป็นผู้ให้ข้อคิด แนวทาง และคำปรึกษา ตลอดจนเป็นผู้ตรวจทานแก้ไข ทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วง ขอขอบพระคุณอาจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์ เป็นอย่างสูงที่ให้ความเมตตา ช่วยเหลือ รวมทั้งโอกาสและสิ่งที่ดีแก่ผู้วิจัยเสมอมา

ขอขอบพระคุณ รองศาสตราจารย์ ดร.สมชาย ประสิทธิ์จูตระกูล รองศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา ผู้ช่วยศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ ประธานกรรมการและกรรมการสอบวิทยานิพนธ์ ที่ได้กรุณาให้คำแนะนำในการแก้ไขวิทยานิพนธ์ให้มีคุณภาพยิ่งขึ้น และขอขอบพระคุณคณาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัยทุกท่านที่ประสิทธิประสาทความรู้อันมีค่ายิ่งแก่ผู้วิจัย

ท้ายนี้ขอขอบพระคุณ บิดา มารดา ที่เป็นกำลังใจสำคัญ และขอขอบคุณ เพื่อนๆ น้องๆ ทุกคน ที่เปรียบเสมือนแรงผลักดันและให้ความช่วยเหลือในทุกๆ ด้านจนผู้วิจัยสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วง

สารบัญ

หน้า

หน้า

สารบัญตาราง

สารบัญภาพ

ภาพที่ หน้า

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัญหาทางการวิเคราะห์สายของข้อมูล เช่น การวิเคราะห์สายดีเอ็นเอ, การวิเคราะห์โครงสร้างของภาษาธรรมชาติ และกลุ่มปัญหาของการรู้จำรูปแบบ (pattern recognition) ปัญหาเหล่านี้อาศัยวิธีการเรียนรู้เป็นหลัก โดยมีงานวิจัยเกี่ยวข้องหลายด้าน เช่น การเรียนรู้ด้วยแบบจำลองโครงข่ายประสาทเทียม (artificial neural network) การเรียนรู้ด้วยอัลกอริทึมเชิงพันธุกรรม (genetic algorithm) และการเรียนรู้ด้วยการสร้างแบบจำลองการส่งผ่าน (transition model) ซึ่งแต่ละวิธีมีข้อดีข้อเสียที่แตกต่างกัน การเรียนรู้ด้วยโครงข่ายประสาทเทียม และการเรียนรู้ด้วยอัลกอริทึมเชิงพันธุกรรมอาศัยการคำนวณจนได้ออกมาเป็นฟังก์ชั่นทางคณิตศาสตร์ที่สามารถคำนวณหาผลลัพธ์เพื่อแยกแยะข้อมูล การเรียนรู้ด้วยแบบจำลองการส่งผ่านจะเป็นการหากฎเกณฑ์ที่แสดงวากยสัมพันธ์ (syntax) ของสายอักขระ ซึ่งเราสามารถพิจารณาให้อยู่ในรูปของไวยากรณ์ (grammar) ได้ ซึ่งทำให้การทำนายหรือตรวจจับความผิดปรกติของรูปแบบที่เกิดขึ้นสามารถทำได้ง่ายขึ้น และยังสามารถคาดเดาข้อมูลที่ไม่พบเจอมาก่อนได้ การพิจารณาปัญหาเหล่านี้ให้อยู่ในรูปของไวยากรณ์นั้นต้องอาศัยการเรียนรู้จากกลุ่มตัวอย่างของข้อมูลที่เคยเกิดขึ้น วิธีนี้เรียกว่า การอนุมานไวยากรณ์ (grammatical inference)

จากการศึกษาในทางทฤษฎีเกี่ยวกับการเรียนรู้โดยโกลด์ [1] ว่าความสามารถในการเรียนรู้ด้วยการอนุมานจากตัวอย่างที่อยู่ในระดับภาษาเวียนเกิดแบบแจงนับแล้ว (recursively enumerable language) สามารถระบุได้โดยจำกัดจากตัวแทนของภาษา หรือเรียกว่าหลักการจำแนกโดยจำกัด (indentification in limit) ซึ่งต้องอาศัยตัวช่วยในการบอกว่าสายอักขระใดอยู่ในภาษาหรือไม่อยู่ในภาษา สายอักขระใดอยู่ในภาษาเรียกอีกอย่างหนึ่งว่ากลุ่มข้อมูลตัวอย่างบวก (positive example) และสายอักขระใดไม่อยู่ในภาษาเรียกอีกอย่างหนึ่งว่ากลุ่มข้อมูลตัวอย่างลบ (negative example) แต่ไม่ได้ระบุเวลาที่ใช้ในการคำนวณไว้ ต่อมาอิกูรา [2] ได้พิสูจน์เกี่ยวพหุนามที่ใช้ในการคำนวณและข้อมูล(polynomial time and data) ว่าการระบุภาษาในระดับไวยากรณ์ไม่พึ่งบริบท (context-free grammar) ไม่สามารถทำได้ จากข้อมูลตัวอย่างบวก และข้อมูลตัวอย่างลบ ดังนั้นเพื่อที่จะสามารถเรียนรู้รูปแบบที่ซับซ้อนในระดับไวยากรณ์ไม่พึ่งบริบทได้โดยลดความซับซ้อนในการคำนวณให้น้อยลงจากการพิจารณากลุ่มตัวอย่างของข้อมูลที่เกิดขึ้นแล้ว จะช่วยให้ปัญหาของการรู้จำรูปแบบทำได้รวดเร็วและสามารถจดจำรูปแบบได้มากขึ้น

ดังนั้นในงานวิจัยนี้มุ่งเน้นในการพัฒนาและปรับปรุงประสิทธิภาพด้านเวลาของอัลกอริทึมการอนุมานไวยากรณ์ระดับไม่พึ่งบริบท โดยพิจารณาจากตัวอย่างบวกและตัวอย่างลบ

1.2 วัตถุประสงค์ของการวิจัย

นำเสนออัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบทที่ใช้เวลาในการคำนวณน้อยและผลลัพธ์ที่ได้เป็นไวยากรณ์ไม่พึ่งบริบท

1.3 ขอบเขตของการวิจัย

1.3.1 ข้อมูลที่จะใส่เข้ามาต้องเป็นภาษาที่อยู่ในระดับไม่พึ่งบริบทเท่านั้น

1.3.2 ตัวอย่างบวก และตัวอย่างลบต้องมีจำนวนจำกัด

1.3.3 ไวยากรณ์ที่ได้ไม่คำนึงถึงจำนวนกฎที่น้อยที่สุดในการบรรยายภาษา

1.3.4 ความถูกต้องของไวยากรณ์จะขึ้นอยู่กับจำนวนของตัวอย่างบวกและตัวอย่างลบ

1.3.5 อัลกอริทึมที่ได้ต้องใช้ความซับซ้อนเชิงเวลาที่น้อยลงเมื่อเทียบกับอัลกอริทึมการเรียนรู้ไวยากรณ์ไม่พึ่งบริบทที่อาศัยพื้นฐานมาจากแนวคิดของซีวายเค (Incremental learning of context-free grammars based on bottom-up parsing and search) [3]

1.4 ขั้นตอนและวิธีดำเนินการวิจัย

1.4.1 ศึกษาทำความเข้าใจเกี่ยวกับไวยากรณ์ไม่พึ่งบริบท

1.4.2 ศึกษางานวิจัยที่เกี่ยวกับกับการอนุมานไวยากรณ์ไม่พึ่งบริบท

1.4.3 ออกแบบอัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบทโดยเน้นที่สามารถทำงานได้รวดเร็ว

1.4.4 ทดสอบวิธีการที่นำเสนอ

1.4.5 เปรียบเทียบผลที่ได้จากการทดสอบ

1.4.6 สรุปผลการวิจัย พร้อมข้อเสนอแนะ และจัดทำรายงานวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย

ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัยนี้คือ ได้อัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบทที่ใช้เวลาลดลง โดยไวยากรณ์ไม่พึ่งบริบทที่ได้ยังมีความถูกต้องตามข้อมูลที่ได้รับเข้ามา

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้ ผู้วิจัยขอกล่าวถึงความรู้พื้นฐาน ทฤษฎีและงานวิจัยที่เกี่ยวข้องกับงานวิจัยนี้ เริ่มด้วยนิยามของภาษารูปนัย ภาษาสม่ำเสมอ เครื่องจักรแบบจำกัดสถานะ คุณสมบัติของภาษาสม่ำเสมอ ไวยากรณ์ไม่พึ่งบริบท ภาษาไม่พึ่งบริบท การแจงส่วน การเรียนรู้ด้วยการอุปนัย และการอนุมานไวยากรณ์ไม่พึ่งบริบท จะถูกกล่าวถึงเป็นอันดับสุดท้าย ซึ่งอัลกอริทึมการเรียนรู้ไวยากรณ์ไม่พึ่งบริบทที่อาศัยพื้นฐานมาจากแนวคิดของซีวายเค (Incremental learning of context-free grammars based on bottom-up parsing and search) [3] จะถูกกล่าวถึงไว้ในส่วนนี้ด้วย

2.1 ภาษารูปนัย (Formal Languages)

เราให้คำจำกัดความของ ชุดตัวอักษร ตัวอักษรหรืออักขระ คำ และภาษาไว้ดังนี้

นิยามที่ 2.1 ชุดตัวอักษร (alphabet) *หมายถึงเซตจำกัดของสัญลักษณ์ ที่เป็นหน่วยย่อยสุด ไม่สามารถแบ่งแยกได้ นิยมใช้สัญลักษณ์แทนชุดตัวอักษรด้วย* $\sum$ *และเรียกสมาชิกในชุดตัวอักษรว่า อักขระ หรือตัวอักษร (character)*

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

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

นิยามที่ 2.4 ภาษารูปนัย (formal language) *หมายถึง ภาษาที่คำต่างๆ ในภาษามีกฎหรือกติกาที่ชัดเจนในการพิจารณาว่าเป็นสมาชิกในภาษานั้นๆ โดยปราศจากความกำกวม นอกจากนี้ จำนวนของกฎหรือกติกาต้องมีเป็นจำนวนจำกัดด้วย*

นิยามที่ 2.5 *สายอักขระที่ไม่มีอักขระเลยจะถูกเรียกว่าเป็น* สายอักขระว่าง (null-string or empty string) *นิยมใช้สัญลักษณ์แทนด้วย* λ

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

นิยามที่ 2.6 ความยาว (length) *ของสายอักขระ หมายถึงจำนวนของอักขระที่ประกอบอยู่ในสายนั้น*

สำหรับสายอักขระว่าง เราจะถือว่า ความยาวจะมีค่าเป็น 0 เสมอ

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

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

โดยที่ $S^i = \{ x_1x_2x_3...x_i \mid \forall x_j \in S\}$

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 $S^i = \{ x_1x_2x_3...x_i \mid \forall x_j \in S\}$

จากนิยาม เราพอสังเกตได้ว่า สำหรับเซต S ใดๆ จะได้ว่า $S+ \subseteq S^*$

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

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

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

2.2 ภาษาสม่ำเสมอ (Regular Language)

เราสามารถนิยามตัวดำเนินการในการอธิบายภาษาสม่ำเสมอได้ดังนี้

นิยามที่ 2.9 ตัวดำเนินการคลีนสตาร์ (Kleene's star or Kleene's closure) บนตัวอักขระ a *เขียนแทนด้วย a* หมายถึงเซตของสายอักขระที่เกิดจากการเรียงต่อกันของ a จำนวนเท่าใดก็ได้*

$$a^* = \{ \lambda, a, aa, aaa, aaaa, \dots \}$$

นิยามที่ 2.10 ตัวดำเนินการบวก (positive closure) ของตัวอักขระ a และ b *เขียนแทนด้วย a + b หมายถึงเซตของสายอักขระที่เกิดจากการเลือกตัวอักขระ a หรือตัวอักขระ b อย่างใดอย่างหนึ่ง*

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

นิยามที่ 2.11 การบรรยายสม่ำเสมอ (regular expression)

กำหนดให้ Σ เป็นชุดตัวอักษร และการบรรยายสม่ำเสมอหมายถึงข้อกำหนดต่อไปนี้

1. ทุกสมาชิกใน Σ เป็น การบรรยายสม่ำเสมอ

2. อักขระว่าง λ เป็น การบรรยายสม่ำเสมอ

3. สำหรับ x และ y ที่เป็น การบรรยายสม่ำเสมอ แล้วจะได้ว่า (x), x+y, xy และ x เป็น การบรรยายสม่ำเสมอด้วย*

4. ใช้เฉพาะกฎ 3 ข้อข้างต้นเท่านั้นในการสร้าง การบรรยายสม่ำเสมอ

นิยามที่ 2.12 *ภาษาใดที่สามารถเขียนบรรยายได้ด้วย การบรรยายสม่ำเสมอ จะเรียกภาษานั้นว่าเป็น* ภาษาสม่ำเสมอ (regular language)

2.3 เครื่องจักรแบบจำกัดสถานะ (Finite State Machine)

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

นิยามที่ 2.13 เครื่องจักรออโตมาตาแบบจำกัด (finite automaton: FA) *หรือ* เครื่องจักรสถานะแบบจำกัด (finite state machine) *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 states) และ $A \subseteq Q$ *(subset) เสมอ*

δ *เป็นฟังก์ชันการเปลี่ยนสถานะ หรือเรียกว่า ฟังก์ชันการผ่าน(transition function) ที่กำหนดโดย*

$$\delta : Q \times \Sigma \rightarrow Q$$

การทำงานของออโตมาตานั้น จะทำงานโดยการรับข้อมูลนำเข้าที่เป็นสายอักขระ โดยการอ่านข้อมูลทีละอักขระ ตามลำดับ ทั้งนี้เพราะออโตมาตาเป็นเครื่องจักรที่มีการทำงานแบบลำดับ (sequential machine) โดยในตอนเริ่มต้นก่อนการทำงาน ออโตมาตาจะอยู่ในสถานะเริ่มต้น เมื่อข้อมูลนำเข้าถูกอ่านทีละอักขระ ออโตมาตาจะเปลี่ยนสถานะของเครื่องจักรไปตามข้อมูลที่อ่านเข้ามานั้น ซึ่งการเปลี่ยนสถานะนี้ได้ถูกกำหนดไว้แล้วโดย ฟังก์ชันการเปลี่ยนสถานะ จากนั้นเริ่มต้นจากสถานะที่เป็นอยู่ ออโตมาตาจะดำเนินการอ่านต่อไปทีละอักขระพร้อมกับการพิจารณาเปลี่ยนสถานะไปตามฟังก์ชัน จนกระทั่งอักขระสุดท้ายถูกอ่านไปเรียบร้อยแล้ว ออโตมาตาจะหยุดทำงาน สถานะสุดท้ายที่ออโตมาตาหยุดการทำงานจะมีผลต่อคำตอบ เพราะถ้าออโตมาตาหยุดที่สถานะที่เป็นสถานะของการยอมรับ เราจะเรียกว่า ออโตมาตานั้นยอมรับข้อมูลนำเข้า ไม่เช่นนั้นจะเรียกว่า ออโตมาตานั้นปฏิเสธข้อมูลนำเข้า ดังนิยามต่อไปนี้

นิยามที่ 2.14 *สายอักขระที่ทำให้ออโตมาตาหยุดที่สถานะยอมรับ จะถูกเรียกว่า* สายอักขระนั้นถูกยอมรับโดยออโตมาตา (accepted by automata) *ไม่เช่นนั้นจะถูกเรียกว่า* ถูกปฏิเสธโดยออโตมาตา (rejected by automata)

2.4 คุณสมบัติของภาษาสม่ำเสมอ

ในที่นี้เราจะกล่าวถึงคุณสมบัติบางประการของภาษาสม่ำเสมอ เราทราบว่า ภาษาที่เป็นภาษาสม่ำเสมอนั้น จะต้องมีออโตมาตาที่ยอมรับ ซึ่งออโตมาตามีจำนวนสถานะที่เป็นจำนวนจำกัด แต่ภาษาสม่ำเสมอนั้นอาจมีจำนวนเป็นอนันต์ได้ ทำให้เห็นว่าสำหรับคำใดก็ตามที่อยู่ในภาษา ถ้าคำนั้นมีความยาวเกินกว่าหรือเท่ากับจำนวนสถานะในออโตมาตาแล้ว จะต้องมีการวงจร (circuit) เกิดขึ้นในเส้นทาง (path) ที่ยอมรับคำนั้น คุณสมบัติที่ว่านี้เองที่ทุกภาษาสม่ำเสมอต้องมี

ทฤษฎีบทที่ 2.1 *สำหรับภาษา L ที่เป็นภาษาสม่ำเสมอแล้ว และยอมรับได้โดยออโตมาตา ที่มีจำนวน สถานะเท่ากับ n สำหรับคำ x ในภาษา L ที่* $|x| \geq n$ *แล้ว x สามารถที่จะเขียนได้เป็น x = uvw สำหรับบาง u, v, และ w ที่สอดคล้องกับ*

$|uv| \geq n$

$|v| > 0$

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

2.5 ไวยากรณ์ไม่พึ่งบริบท (Context-Free Grammar)

ไวยากรณ์ไม่พึ่งบริบท (context-free grammar) เป็นวิธีการบรรยายภาษาด้วยกฎที่ใช้ในการสร้างที่มีจำนวนจำกัด ด้วยวิธีการสร้างแบบวนซ้ำ (recursive method) ซึ่งเป็นกฎเพียงกฎเดียวที่สามารถจะทำการนำมาใช้ได้หลายๆ ครั้ง

นิยามที่ 2.15 ไวยากรณ์ไม่พึ่งบริบท (context-free grammar) *หมายถึงการบรรยายภาษาด้วยกฎไวยากรณ์ ซึ่งประกอบด้วยส่วนสำคัญสี่ส่วน ดังนี้*

$$G = (V, T, P, S)$$

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

T *ชุดของตัวอักษรจำกัด (finite set of alphabet or terminal) เซตที่มีสมาชิกไม่ซ้ำกับเซตของตัวแปร*

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

P *เซตจำกัดของกฎไวยากรณ์ (finite set of grammar rules or production) เป็นเซ็ตของกฎซึ่งอยู่ในรูปของ*

$$A \rightarrow a$$

เมื่อ $A \in V$ *และ* $a \in (V \cup T)^*$

นิยามที่ 2.16 ภาษาไม่พึ่งบริบท (context-free language) *คือภาษาที่สัมพันธ์กับไวยากรณ์ไม่พึ่งบริบท G หมายถึงเซตของสายอักขระที่มีคุณสมบัติดังนี้*

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

นิยามที่ 2.17 *สายอักขระ* $x \in (V \cup \Sigma)^*$ *ใดๆ ซึ่งมีตัวแปรปนกับตัวอักษร นั้นเราจะเรียกว่า* รูปประโยค (sentential form)

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

G = (V, T, P, S)

V = {S}

W = {a,b}

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

ตัวอย่างการแปลง (derivation) ให้ได้ aaabbb ที่เป็นสมาชิกของ L

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

นิยามที่ 2.18 *ไวยากรณ์ไม่พึ่งบริบท Gใดๆ เป็น*ไวยากรณ์ที่ไม่มีความกำกวม (unambiguous grammar) *ก็ต่อเมื่อทุกสายอักขระที่เป็นสมาชิกของภาษา L(G) สามารถทำการแปลงตัวแปรจากทางซ้ายสุดก่อนได้เพียงแบบเดียว แต่ถ้ามีคำที่เป็นสมาชิกของภาษา L(G) สามารถทำการแปลงตัวแปรจากทางซ้ายสุดได้มากกว่าหนึ่งแบบ โดยใช้ลำดับการแปลงที่แตกต่างกันและเรียกไวยากรณ์นี้ว่า* ไวยากรณ์ที่มีความกำกวม (ambiguous grammar)

ตัวอย่างเช่น
จากกฎไวยากรณ์ของภาษากำหนดมาให้ดังนี้

$$S \rightarrow S + S \mid a$$

โดยที่ T = {+, a}

จะเห็นว่า a+a+a สามารถทำการแปลงตัวแปรจากทางซ้ายสุดก่อนได้สองแบบคือ

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

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

2.6 การแจงส่วน (Parsing)

การแจงส่วน คือ การหาการแปลงของไวยากรณ์ G จากสัญลักษณ์เริ่มต้นของไวยากรณ์ S ไปยังสายอักขระ ซึ่งถ้าสามารถหาการแปลงจากสัญลักษณ์เริ่มต้นของไวยากรณ์ ไปยังสายอักขระได้ แสดงว่าสายอักขระนั้นเป็นสมาชิกของ L(G) อัลกอริทึมในการตรวจสอบคำใดๆ ว่าเป็นสมาชิกของ L(G) หรือไม่ โดยที่ L(G) นิยามจากไวยากรณ์ G อัลกอริทึมที่นิยมนำมาใช้ในการแจงส่วน เพื่อตรวจข้อมูลที่เข้ามากับไวยากรณ์ที่สร้างขึ้นในระดับไวยากรณ์ไม่พึ่งบริบท คือ อัลกอริทึมการแจงส่วนของ Cocke-Younger-Kasami (CYK)[4] วิธีของซีวายเค ใช้การแจงส่วนแบบล่างขึ้นบน(bottom-up parsing) และการทำงานของอัลกอริทึมเป็นแบบโปรแกรมเชิงพลวัต (dynamic programming) ในการคำนวณซึ่งจะลดการแจงส่วนคำย่อยที่เคยแจงส่วนไปแล้ว ความซับซ้อนเชิงเวลา (time complexity) ที่ใช้คำนวณคำใดๆ อยู่ในรูป $O(n^3)$ โดยที่ n เป็นความยาวของคำที่แจงส่วน และอีกอัลกอริทึมการแจงส่วนของ Earley[5] ซึ่งใช้โปรแกรมเชิงพลวัตเช่นกัน มีความซับซ้อนเชิงเวลาในกรณีเลวร้ายที่สุด อยู่ในรูป $O(n^3)$ เช่นกัน แต่ถ้าเป็นไวยากรณ์ที่ไม่มีความกำกวม จะอยู่ในรูป $O(n^2)$

2.7 การเรียนรู้ด้วยการอุปนัย

การเรียนรู้ด้วยการอุปนัย หมายถึง การเรียนรู้จากข้อมูลตัวอย่างเพื่อให้ได้มาซึ่งกฎเกณฑ์ทั่วไปของชุดข้อมูลตัวอย่างนั้น[6] ตัวอย่างเช่น การเรียนรู้จากตัวอย่างชุดข้อมูล aaabb, aabb, aabbbb , aaaabbbb, aaabbbb อาจจะสรุปได้ว่า กฎเกณฑ์ที่อธิบายสภาพข้อมูลนี้คือ สายอักขระที่ขึ้นต้นด้วย a ไม่จำกัดจำนวน และตามด้วย b ไม่จำกัดจำนวน ซึ่งคำ

จำกัดความของการเรียนรู้ด้วยการอุปนัยนี้ อาจจะไม่เหมือนคำว่า การเรียนรู้ ในความหมายทั่วไปตรงที่ การเรียนรู้ด้วยการอุปนัยจะสนใจแต่กรณีของการได้มาซึ่งกฎเกณฑ์ทั่วไปของข้อมูล โดยที่ยังไม่ได้คำนึงถึงคำตอบว่าสิ่งใดอยู่หรือไม่อยู่ในกลุ่มข้อมูลตัวอย่าง แต่การเรียนรู้ในความหมายทั่วไป จะสนใจในแง่ของการตอบคำถามว่าใช่หรือไม่ มากกว่ากฎเกณฑ์ที่อธิบายสภาพของข้อมูล ในการเรียนรู้ด้วยอุปนัยนั้นจะระบุปัญหาของการเรียนรู้ด้วยการอุปนัยดังนี้

1. ชั้นของฟังก์ชันหรือชั้นของภาษาที่จะพิจารณา ตัวอย่างเช่น กฎเกณฑ์ที่ได้จากการเรียนรู้อยู่ในชั้นของภาษาไวษากรณ์สม่ำเสมอ

2. สมมติฐานที่เป็นไปได้ (hypothesis space) หมายถึง ขอบเขตของสมมติฐานที่ได้จากการเรียนรู้ที่เป็นไปได้ ตัวอย่างเช่น ในการเรียนรู้ภาษาสม่ำเสมอ เราจะออกแบบกฎให้อยู่ในรูปของออโตมาตา หรือ ไวยากรณ์สม่ำเสมอ ต้องคำนึงถึงสมมติฐานเป็นไปได้ทั้งหมด

3. ตัวอย่างที่ยอมรับได้ในการในการแสดงออก (admissible presentation) ยกตัวอย่างเช่น ข้อมูลตัวอย่างประกอบตัวด้วยข้อมูลตัวอย่างที่อยู่ในภาษา และข้อมูลตัวอย่างข้อมูลที่ไม่อยู่ในภาษา

4. วิธีการพิจารณาว่าสมมติฐานใดที่เป็นไปได้ ตัวอย่างเช่น อัลกอริทึม

5. ข้อกำหนดขอบเขตความสำเร็จของการเรียนรู้ (criterion of success) เช่น การยอมรับว่าการเรียนรู้สำเร็จเมื่อพบว่าสามารถระบุตัวอย่างที่อยู่ในภาษาได้

2.8 การอนุมานไวยากรณ์ไม่พึ่งบริบท (Context-Free Grammar Inference)

การอนุมานไวยากรณ์ไม่พึ่งบริบท คือ ปัญหาการหาไวยากรณ์ไม่พึ่งบริบท ที่สามารถระบุภาษาในระดับไวยากรณ์ไม่พึ่งบริบท จากกลุ่มข้อมูลตัวอย่างที่เป็นสายอักขระในภาษา ซึ่งอาจจะประกอบไปด้วยกลุ่มข้อมูลตัวอย่างที่อยู่ในภาษาเรียกว่ากลุ่มข้อมูลตัวอย่างบวก และ/หรือกลุ่มข้อมูลตัวอย่างที่ไม่อยู่ในภาษา เรียกว่ากลุ่มข้อมูลตัวอย่างลบ และการระบุความสำเร็จในการอนุมานไวยากรณ์ ที่นิยมนำมาเป็นตัววัดความสำเร็จ คือ หลักการจำแนกโดยจำกัด (language identification in limit) ของโกลด์

2.8.1 หลักการจำแนกภาษาภายในจำกัด

ตัวอย่างมุมมองของการเรียนรู้ด้วยหลักการจำแนกภายในจำกัด จากเกมเดาตัวเลขว่าต่อจากลำดับ 1, 3, 5... ควรจะเป็นอะไร จากการเรียนรู้ด้วย 1, 3, 5 อาจจะเดาโดยตั้งสมมติฐานแรกว่าจะต้องเป็นชุดของเลขคี่ หรือ 2n+1 ดังนั้นเมื่อตัวต่อไปเป็นเลข 7 เข้ามาสรุปว่าฟังก์ชันเลขคี่ยังคงถูกต้องอยู่ จากนั้นเมื่อตัวเลขถัดไปเข้ามาเป็น 11 ปรากฏว่าฟังก์ชั่นเลขคี่ไม่ถูกต้อง จึงจำเป็นต้องเปลี่ยนสมมติฐานด้วยการเดาเป็นเลขของจำนวนเฉพาะ ผลปรากฏว่าเลขที่เข้ามาใหม่ตัวต่อไปทั้งหมดเป็นลำดับดังนี้ คือ 13, 17, 19, 23 ซึ่งถูกต้องทั้งหมด ดังนั้นจึงกล่าวได้ว่าจากตัวเลขที่เข้ามา ฟังก์ชันเลขจำนวนเฉพาะคือฟังก์ชัน หรือสมมติฐานการ

เรียนรู้ที่ถูกต้อง ตราบใดที่ยังไม่มีชุดตัวเลขที่ทำให้สมมติฐานไม่เป็นจริง สรุปว่าการเรียนรู้ด้วยการจำแนกภายในจำกัด หมายถึง การกำหนดขอบเขตความสำเร็จของการเรียนรู้ด้วยหลักการว่า เมื่อมีการเรียนรู้ไปได้ระยะหนึ่งสมมติฐานที่ถูกต้องนั้นจะไม่เปลี่ยนแปลงและจำนวนครั้งในการเปลี่ยนสมมติฐานนั้นรับประกันได้ว่ามีจำกัด กำหนดให้ G_t แทนฟังก์ชันที่ได้จากการเรียนรู้เมื่อรับสายอักขระลำดับที่ t จากกลุ่มข้อมูลตัวอย่างแล้วเรายอมรับว่าการเรียนรู้สำเร็จ ณ ลำดับที่ t เมื่อพบว่า

$$G_t = G_{t+1} = G_{t+2} = G_{t+3} = G_{t+4} = \ldots$$

ลำพังเพียงการเรียนรู้ด้วยการจำแนกภายในจำกัดนั้นยังขาดประสิทธิภาพ จึงได้มีการนำเสนอหลักเกณฑ์เพื่อรับประกันประสิทธิภาพ โดยกล่าวว่าการเรียนรู้ใดๆนั้นนอกจากจะสามารถจำแนกเอกลักษณ์ได้แล้ว ยังจะต้องใช้เวลาในการเรียนรู้ไม่เกินฟังก์ชันพหุนาม รวมถึงอัตราการเติบโตของข้อมูลที่ได้จากการเรียนรู้จะต้องไม่เกินฟังก์ชันพหุนาม จากความรู้ในปัจจุบัน การอนุมานไวยากรณ์สม่ำเสมอ(regular grammar) ประสบผลสำเร็จในการจำแนกภายในจำกัดเชิงเวลาและเชิงข้อมูลในฟังก์ชันพหุนาม[7] แต่ไม่สามารถหาออโตมาตาแบบจำกัดเชิงกำหนด (deterministic finite automata) ซึ่งเทียบเท่ากับไวยากรณ์สม่ำเสมอที่เล็กที่สุดได้ในเวลาฟังก์ชันพหุนาม[8] ส่วนการอนุมานไวยากรณ์ไม่พึ่งบริบทในทางทฤษฎีแล้วไม่สามารถจำแนกภายในจำกัดเชิงเวลาและข้อมูลได้ [2]

นอกจากหลักการจำแนกโดยจำกัดแล้วยังมีรูปแบบการวัดความสำเร็จของการเรียนรู้อีกหลายอย่าง เช่น การเรียนรู้ด้วยการประมาณความถูกต้องโดยความน่าจะเป็น (probably approximately correct : PAC) หรือ พีเอซี คือการเรียนรู้โดยการคำนวณเพื่อหาค่าความน่าจะเป็นให้ตรงตามข้อมูลตัวอย่าง โดยมีหลักเกณฑ์ว่า สามารถยอมรับค่าความน่าจะเป็นที่มีความคลาดเคลื่อนระดับหนึ่งได้ ถ้าความคลาดเคลื่อนไม่เกินค่าใดค่าหนึ่ง ซึ่งค่านี้อาจจะถูกกำหนดเอาไว้ล่วงหน้า และเมื่อใดก็ตามที่มีดำเนินการเรียนรู้จนได้ความคลาดเคลื่อนไม่มากไปกว่าที่กำหนดแล้ว เราสามารถยอมรับได้ว่าเรียนรู้สำเร็จ นอกจากนี้ยังมีข้อจำกัดคือเวลาในการเรียนรู้และขนาดของแบบจำลองจะต้องไม่เติบโตเกินฟังก์ชั่นพหุนาม จากความรู้ในปัจจุบันยังไม่มีข้อที่พิสูจน์ได้ว่า เมื่อนำมาประยุกต์ใช้กับการเรียนรู้ไวยากรณ์แล้ว จะสามารถทำนายได้ถูกต้องตามหลักการหรือไม่ มีแต่แนวโน้มว่าไม่สามารถที่จะนำมาใช้ได้เนื่องจากมีการพิสูจน์ว่าปัญหาในการทำนายข้อมูลมีความยากเท่ากับบางปัญหาในการเข้ารหัส (cryptographic problem) [9]

ดังนั้นงานวิจัยส่วนใหญ่จึงมุ่งไปสู่หลักการจำแนกโดยจำกัดโดยหาชั้นย่อยของภาษาไวยากรณ์ไม่พึ่งบริบท ซึ่งมีหลายงานที่ประสบความสำเร็จในหลักการจำแนกโดยจำกัดเชิงเวลาและข้อมูลในฟังก์ชันพหุนาม โดยได้นิยามคลาสของภาษาขึ้นมาใหม่ซึ่งเป็นคลาสย่อยของภาษาไวยากรณ์ไม่พึ่งบริบท ซึ่งได้พิสูจน์แล้วว่าสามารถจำแนกโดยจำกัดเชิงเวลาและข้อมูล เช่น

ไวยากรณ์คู่เชิงเส้น (even linear grammar) [10], ไวยากรณ์เชิงเส้นเชิงกำหนด (deterministic linear grammar) [11], ไวยากรณ์อย่างง่ายมาก (very simple grammar) [12]

2.9 งานวิจัยที่เกี่ยวข้อง

2.9.1 Incremental learning of context-free grammars based on bottom-up parsing and search

ในปี 2005 นากามุระ[3] ได้เสนออัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบท ใช้การพิจารณาจากตัวอย่างบวกและตัวอย่างลบ โดยอาศัยอัลกอริทึมการแจงส่วน CYK เป็นเครื่องมือในการพิจารณาสร้างกฎ มีข้อจำกัดคือไม่สามารถอนุมานไวยากรณ์ไม่พึ่งบริบทที่มีตัวแปรมากกว่า 12 ตัวแปรได้ เนื่องจากปัญหาด้านเวลาที่ใช้ ซึ่งรายละเอียดของอัลกอริทึม จะแสดงได้รูปที่ 2.1 และ 2.2

Top-Level Search Algorithm

Input S_P: an ordered set of positive sample strings;
S_N: an ordered set of negative sample strings;
P_0: an initial set of rules; and $K_{\max}$: the limit of the number of rules.

Output A set P of rules such that all the strings in S_P are derived from P but no string in S_N is derived from P.

Procedure

Step 1: Initialize variables $P \leftarrow P_0$ (the set of rules),
$N \leftarrow \{S\} \cup \{$the set of nonterminal symbols in $P_0\}$, and
$K \leftarrow |P_0|$ (the limit of the number of rules).

Step 2: For each $w \in S_P$, iterate the following operations.
(1) Find a set of rules by calling inductive CYK algorithm with the inputs w, P, N and K. Assign the results to P and N.
(2) For each $v \in S_N$, test whether v is derived from P by CYK algorithm. If there is a string v derived from P, then backtrack to the previous choice point.
If no set of rules is obtained, then
(1) If $K \geq K_{\max}$, terminate (no set of rules is found within the limit).
(2) Otherwise, add 1 to K and restart Step 2.

Step 3: Output the result P.
For finding multiple solutions, backtrack to the previous choice point.
Otherwise, terminate.

รูปที่ 2.1 อัลกอริทึมหลักจากงานวิจัย [3]

Inductive CYK Algorithm

Input w: a string, P_0: a set of rules; N: a set of nonterminal symbols; and K: an integer (the limit of the number of rules).

Output A set of rules from which w is derived and a set of nonterminal symbols in the rules.

Procedure Initialize the variables $P_1 \leftarrow \emptyset$ (the set of generated rules); and $TS \leftarrow \emptyset$ (the test set).

Repeat Steps 1 and 2 until w is derived from $P_0 \cup P_1$.

Step 1: (Test whether w is derived from $P_0 \cup P_1$ by CYK algorithm, and at the same time generate a test set TS used in Step2.)

(1) Consider w as the string $a_1 a_2 \cdots a_n$. Initialize a 2-dimensional array T by $T[i,1] = \{a_i\}$ for all $1 \leq i \leq n$.

(2) (Find all elements $T[i,j]$ of T such that $A \Rightarrow^* a_i \cdots a_{i+j-1}$ for all $A \in T[i,j]$.) Iterate the following processes for $2 \leq j \leq n$ and for $1 \leq i \leq n-j+1$.

(a) $T[i,j] \leftarrow \emptyset$;

(b) For all k $(1 \leq k \leq j-1)$, $\beta \in T[i,k]$, and $\gamma \in T[i+k, j-k]$,

(i) $TS \leftarrow TS \cup \{(\beta, \gamma)\}$;

(ii) if $(A \rightarrow \beta\gamma) \in P_0 \cup P_1$ then
(for generating unambiguous grammars, if $A \in T[i,j]$ then backtrack to the previous choice point(failure);)
$T[i,j] \leftarrow T[i,j] \cup \{A\}$;

(3) If $S \in T[1,n]$ then return the values $P_0 \cup P_1$ and N (success).

(4) If $|P_0 \cup P_1| \geq K$, then backtrack to previous choice point (failure).

Step 2: (Generate a rule $A \rightarrow \beta\gamma$ and add it to P_1, where (β, γ) is a pair contained in the test set TS.)

(1) † Nondeterministically select a pair $(\beta, \gamma) \in TS$.

(2) † Nondeterministically select a nonterminal symbol $A \in N$ such that $(A \rightarrow \beta\gamma) \notin P_0 \cup P_1$, or generate a new nonterminal symbol A and add it to N by $N \leftarrow N \cup \{A\}$.

(3) $P_1 \leftarrow P_1 \cup \{(A \rightarrow \beta\gamma)\}$.

† Choice points for backtracking.

รูปที่ 2.2 อัลกอริทึมการหากฎจากตารางซีวายเคจากงานวิจัย [3]

หลักการทำงานของอัลกอริทึมใช้การหาเซ็ตของกฎจากตัวแปรที่สามารถแจงส่วนตัวอย่างบวกได้และแต่ไม่สามารถแจงส่วนตัวอย่างลบได้ โดยอาศัยการกำหนดตัวแปรจากตารางการแจงส่วนของซีวายเค โดยเริ่มจากตัวแปรเริ่มต้น S เพียง 1 ตัว เมื่อไม่สามารถสร้างกฎโดยใช้จำนวนตัวแปรที่มีอยู่ได้ จึงเพิ่มจำนวนตัวแปรขึ้นทีละตัว และกฎที่ได้จะอยู่ในบรรทัดฐานของชอมสกี (chomsky normal form) และ เวลาการทำงานของอัลกอริทึมจะเพิ่มขึ้นเป็นแบบชี้กำลัง ตามจำนวนของตัวแปรที่เพิ่มขึ้น แต่จะมีค่ากำหนดว่าจะไม่ทำการสร้างกฎมากไปกว่าค่า K เมื่อได้กฎที่สามารถสร้างตัวอย่างบวกได้หนึ่งตัวจะนำกฎที่ได้ไปตรวจสอบกับตัวอย่าง

ลบทั้งหมด ถ้าตรวจสอบกับตัวอย่างลบมีตัวอย่างลบที่สามารถสร้างได้จะทำการย้อนรอย (backtracking) ไปยังจุดทางเลือก (choice points)

2.9.2 Ga-based learning of context-free grammars using tabular representations

ในปี 2005 ซากากิบารา [13,14] ได้เสนออัลกอริทึมในการอนุมานไวยากรณ์ไม่พึ่งบริบท ใช้การพิจารณาจากตัวอย่างบวกและตัวอย่างลบ และใช้อัลกอริทึมเชิงพันธุกรรมเข้ามาค้นหาเซ็ตของตัวแปรที่เป็นไปได้ โดยหาจากโครงสร้างต้นไม้ที่ได้จากตารางการแจงส่วนของซีวายเค คล้ายกับงานของนากามุระ ซึ่งการทำงานมีความไม่แน่นอนเนื่องจากใช้หลักการสุ่ม ในการหาประชากรใหม่ทำให้ไวยากรณ์ที่ได้จากอัลกอริทึมแต่ละครั้งไม่เหมือนกัน

2.9.3 Learning context-free grammars with a simplicity bias

ในปี 2000 แลงเลย์[15] ได้เสนออัลกอริทึมในการอนุมานไวยากรณ์ไม่พึ่งบริบท โดยพิจารณาจากตัวอย่างบวกเพียงอย่างเดียว และไวยากรณ์ที่ใช้ทดสอบจะเป็นไวยากรณ์ที่ใช้ในภาษาอังกฤษ ซึ่งจะมีตัวดำเนินการอยู่สองแบบ คือ การสร้างตัวแปรตัวใหม่ และการรวมตัวแปรเข้าด้วยกัน การสร้างตัวแปรตัวใหม่ทำโดยหาจากกลุ่มของตัวแปรจากกฎทางขวาที่เกิดขึ้นบ่อยนำมาสร้างเป็นตัวแปรใหม่ รูปแบบการพิจารณาการสร้าตัวแปรใหม่ทำได้ดังนี้

กำหนดให้มีกฎ

NP $\longrightarrow$ ART ADJ NOUN

NP $\longrightarrow$ ART ADJ ADJ NOUN

จะสร้างตัวแปรใหม่ได้เป็น

NP $\longrightarrow$ ART AP1

NP $\longrightarrow$ ART ADJ AP1

AP1 $\longrightarrow$ ADJ NOUN

ส่วนการรวมตัวแปร จะทำการพิจารณาดังนี้

กำหนดให้มีกฎ

NP $\longrightarrow$ ART AP1

NP $\longrightarrow$ ART AP2

AP1 $\longrightarrow$ ADJ NOUN

AP2 $\longrightarrow$ ADJ AP1

จะทำการรวมตัวแปรได้เป็น

NP $\rightarrow$ ART AP1

AP1 $\rightarrow$ ADJ NOUN

AP1 $\rightarrow$ ADJ AP1

ซึ่งอัลกอริทึมนี้จะมีตัววัดว่าสมควรจะรวมตัวแปรใดหรือสร้างตัวแปรใดก่อนจากหลักการของฟังก์ชั่นที่หาการบรรยายด้วยความยาวน้อยสั้นสุด (minimal description length : MDL) ซึ่งไม่ได้อธิบายรายละเอียดเอาไว้ หลักการทำงานของอัลกอริทึมจะเริ่มจากการหาการรวมตัวแปรที่ดีที่สุดโดยวัดจากฟังก์ชั่นเอ็มดีแอล จนไม่สามารถรวมตัวแปรได้แล้วจึง เปลี่ยนไปหาการสร้างตัวแปรที่ดีที่สุด โดยวัดจากฟังก์ชั่นเอ็มดีแอล เช่นกัน เมื่อไม่สามารถหาการสร้างตัวแปรใหม่ได้จะเปลี่ยนไปทำการรวมตัวแปรอีกครั้ง ทำไปจนกว่าไม่สามารถหาไวยากรณ์ที่ดีขึ้นได้แล้ว จึงจบการทำงาน ซึ่งข้อเสียของอัลกอริทึมนี้คือการใช้แต่ตัวอย่างบวกเพียงอย่างเดียวทำให้ได้ไวยากรณ์ไวยากรณ์ที่มีความกว้างมากเกินไป

2.9.4 LARS: A learning algorithm for rewriting systems

ในปี 2006 อิกูร่าได้เสนออัลกอริทึมแอลเออาร์เอส (LARS)[16] ใช้การอธิบายภาษาด้วยกฎในรูปแบบการแทนสายอักขระย่อยด้วยอักสายอักขระย่อยที่มีความยาวสั้นกว่า(rewriting systems) ซึ่งการยอมรับของการแทนสายอักขระย่อยด้วยสายอักขระย่อยที่สั้นกว่า คือ จะมีสายอักขระหนึ่งเป็นสายอักขระที่สั้นที่สุดซึ่งไม่สามารถแทนได้สายอักขระใดได้ การพิจารณาว่าสายอักขระใดอยู่ในภาษาหรือไม่ จะพิจารณาจากการแทนสายอักขระย่อยด้วยกฎที่มีอยู่ไปเรื่อยๆจนไม่สามารถแทนได้ และพิจารณาสายอักขระสุดท้ายว่าอยู่เป็นตัวเดียวกับที่กำหนดไว้หรือไม่ ถ้าใช่แสดงว่าสายอักขระนั้นอยู่ในภาษา ถ้าไม่ใช่แสดงว่าสายอักขระนั้นไม่อยู่ในภาษา หลักการทำงานของอัลกอริทึมคือ สร้างกฎทั้งหมดที่เป็นไปได้ โดยพิจารณาการหาสายอักขระย่อยของตัวอย่างบวก นำมาแทนด้วยสายอักขระย่อยที่มีความยาวสั้นกว่า ซึ่งผลที่ได้สามารถอนุมานได้บางส่วนของภาษาสม่ำเสมอและภาษาไม่พึ่งบริบทได้ แต่มีความซับซ้อนเชิงเวลาอยู่ในฟังก์ชั่นพหุนามของความยาวรวมของตัวอย่างบวกและความยาวรวมของตัวอย่างลบ

บทที่ 3

อัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบท

งานวิจัยนี้สนใจปัญหาในการอนุมานไวยากรณ์ของภาษารูปนัยด้วยการเรียนรู้จากตัวอย่างของสมาชิกในภาษาอันได้แก่ ประโยคหรือคำ ตามแต่ความซับซ้อนของภาษานั้นๆ สมาชิกเหล่านี้จะถูกเรียกว่าเป็น ตัวอย่างบวกสำหรับการเรียนรู้ แต่เพื่อให้ได้ภาษาที่ใกล้เคียงความเป็นจริงมากที่สุด งานวิจัยนี้จึงสนใจนำตัวอย่างลบ ซึ่งก็คือสายอักขระที่ไม่ได้เป็นสมาชิกในภาษานั้นมาประกอบการพิจารณาด้วย สำหรับรายละเอียดของบทนี้จะเริ่มต้นจากการอธิบายและนิยามปัญหาของงานวิจัย จากนั้นจะนำเสนอแนวคิดในงานวิจัยซึ่งจะรวมถึงกระบวนการในการแก้ปัญหา อัลกอริทึมของการอนุมานไวยากรณ์ และเพื่อให้ง่ายต่อความเข้าใจ จะมีการยกตัวอย่างประกอบการทำงานซึ่งสามารถช่วยให้เห็นการทำงานในแต่ละขั้นตอนได้ชัดเจนยิ่งขึ้น

3.1 รูปแบบของปัญหา

ทั้งนี้เนื่องจากภาษารูปนัยตามทฤษฎีของภาษานั้น ความซับซ้อนของภาษาสามารถแบ่งออกได้เป็นหลายระดับชั้น โดยความซับซ้อนเหล่านั้นจะแสดงให้เห็นได้จากคุณสมบัติของไวยากรณ์ที่ใช้ในการอธิบายภาษา งานวิจัยนี้มีขอบเขตของการเรียนรู้ที่มุ่งเน้นไปที่ภาษาที่มีความซับซ้อนไม่เกินไปกว่า ภาษาไม่พึ่งบริบท ดังนั้นในการอนุมานไวยากรณ์ เราจะใช้ไวยากรณ์ไม่พึ่งบริบท

ตัวอย่างบวกที่นำมาใช้ในการอนุมาณไวยากรณ์ จะใช้เป็นข้อมูลสำหรับการเรียนรู้รูปแบบที่เกิดขึ้นภายในตัวอย่างแต่ละตัวอย่าง งานวิจัยนี้จะทำการสกัดรูปแบบที่เกิดขึ้นโดยพิจารณาการเกิดซ้ำของรูปแบบไปพร้อมกันด้วย แต่ทั้งนี้เพื่อไม่ให้การอนุมานไวยากรณ์ได้ไวยากรณ์ที่กว้างจนเกินไป ตัวอย่างลบจึงถูกนำมาพิจารณาประกอบการอนุมานไวยากรณ์ด้วย ข้อกำหนดสำคัญคือ ไวยากรณ์ที่ได้ต้องไม่สามารถแจงส่วนตัวอย่างลบได้เลยแม้แต่ตัวอย่างเดียว แต่ต้องสามารถแจงส่วนตัวอย่างบวกได้ทั้งหมด

งานวิจัยนี้ไม่ได้สนใจว่า ไวยากรณ์ที่ได้จากการอนุมานซึ่งเป็นไวยากรณ์ไม่พึ่งบริบทไม่จำเป็นต้องเป็นไวยากรณ์ที่มีจำนวนกฎไวยากรณ์และจำนวนตัวแปรน้อยที่สุดเท่าที่จะทำได้สำหรับเวลาที่ใช้ในการอนุมานไวยากรณ์นั้น ดังปรากฏในงานวิจัย [2] ไว้แล้วว่า ชั้นของภาษาในระดับไวยากรณ์ไม่พึ่งบริบทไม่สามารถจำแนกโดยจำกัดเชิงเวลาและเชิงข้อมูลในฟังก์ชันพหุนามได้

เกณฑ์ที่ใช้ในการประเมินผลและวัดประสิทธิภาพของอัลกอริทึมการอนุมานไวยากรณ์ของภาษาไม่พึ่งบริบทนี้คือ ปริมาณของตัวอย่างบวกที่ใช้ในการลู่เข้าของการอนุมานไวยากรณ์ไม่พึ่งบริบทที่ถูกต้อง

จากที่กล่าวมานี้สามารถกำหนดปัญหาของงานวิจัยได้เป็นข้อสรุปดังต่อไปนี้

ปัญหาของงานวิจัย

กำหนดให้

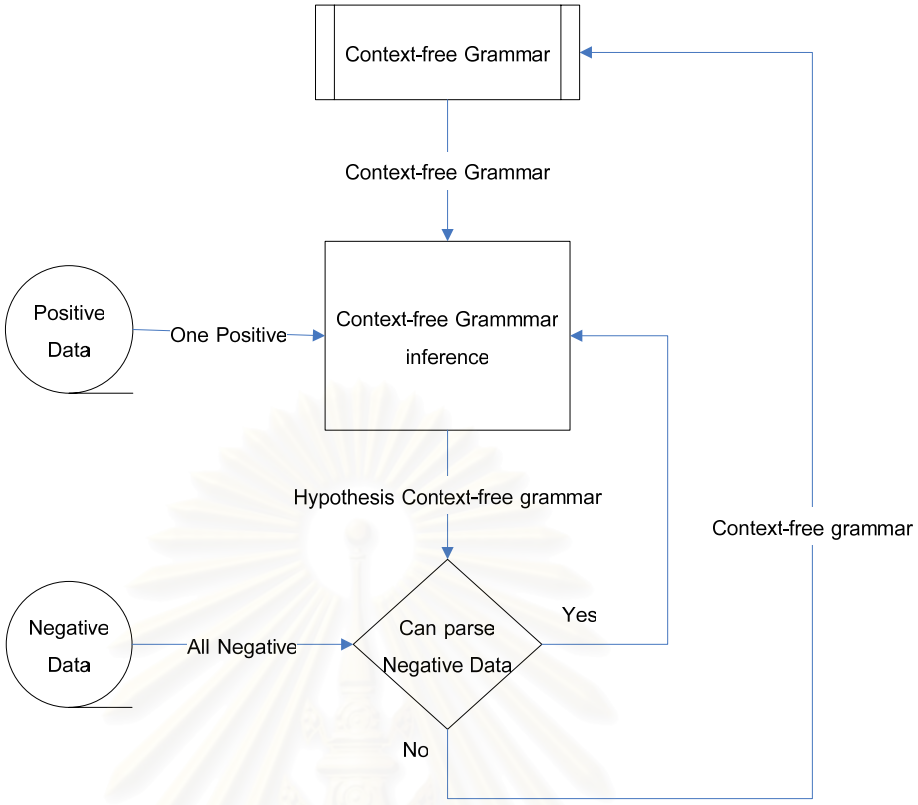
T	เป็นเซตจำกัดของอักขระ
a, b, c, …	เป็นอักขระที่เป็นสมาชิกของเซตจำกัดของอักขระ T โดยอักขระมีจำนวนรูปแบบที่แตกต่างกันเป็นจำนวนจำกัด
S_P	เป็นเซตจำกัดของตัวอย่างบวก โดยตัวอย่างเหล่านั้นจะต้องมีความยาว (ขอบเขต) ที่จำกัด และอักขระทั้งหมดต้องมาจากเซตของอักขระ T
S_N	เป็นเซตจำกัดของตัวอย่างลบ โดยตัวอย่างเหล่านั้นจะต้องมีความยาว (ขอบเขต) ที่จำกัด และอักขระทั้งหมดต้องมาจากเซตของอักขระ T

การอนุมานไวยากรณ์ G = (V, T, P, S) โดยที่ภาษาของไวยากรณ์ G จะต้องมีคุณสมบัติคือ

$$S_P \subseteq L(G) \text{ และ } S_N \cap L(G) = \varnothing.$$

3.2 การทำงานของอัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบท

การทำงานของอัลกอริทึมสำหรับการอนุมานไวยากรณ์ในงานวิจัยนี้ เริ่มต้นจากการพิจารณาข้อมูลนำเข้าที่เป็นตัวอย่างบวก S_P โดยการพิจารณาทีละตัวอย่าง สำหรับข้อกำหนดเบื้องต้น ข้อมูลตัวอย่างบวกทั้งหมดต้องถูกจัดเรียงลำดับตามความยาวของตัวอย่างก่อนการทำงาน จากนั้นแต่ละครั้งที่พิจารณาตัวอย่างบวก จะมีการอนุมานกฎไวยากรณ์ที่สามารถแจงส่วนตัวอย่างบวกที่กำลังพิจารณาอยู่ และรวมถึงตัวอย่างบวกทั้งหมดที่ได้พิจารณาไปแล้วด้วย ซึ่งการอนุมานกฎไวยากรณ์นี้จะต้องคำนึงถึงกฎไวยากรณ์ที่ได้มาแล้วในรอบก่อนหน้าด้วย เมื่อได้กฎไวยากรณ์ใหม่แล้วจากนั้นต้องพิจารณาตัวอย่างลบใน S_N ว่า ต้องไม่สามารถถูกแจงส่วนได้จากกฎไวยากรณ์ที่เพิ่มขึ้นมานี้ การอนุมานไวยากรณ์สามารถทำได้โดยขั้นตอนตามที่กล่าวมาทั้งหมดนี้ ซึ่งอธิบายได้ดังแสดงในรูปที่ 3.1 ผลลัพธ์ที่ได้จากอัลกอริทึมคือ กฎไวยากรณ์สำหรับการแจงส่วนภาษาที่ครอบคลุมตัวอย่างบวกทั้งหมด แต่ไม่รวมถึงตัวอย่างลบทุกตัวอย่าง



รูปที่ 3.1 การทำงานของอัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบท

3.3 อัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบท

การทำงานของอัลกอริทึมเริ่มจากการรับข้อมูลตัวอย่างบวกมาทีละตัวอย่าง โดยที่แต่ละรอบจะเริ่มจากการตรวจสอบว่าตัวอย่างบวกที่เข้ามาด้วยการแจงส่วนก่อนว่ากฎที่มีอยู่ก่อนหน้านี้ว่าสามารถแจงส่วนได้เป็นตัวอย่างบวกที่เข้ามาได้หรือไม่ ถ้าสามารถทำการแจงส่วนได้ แสดงว่าไม่ต้องทำการเพิ่มหรือเปลี่ยนกฎใดๆ แต่ถ้าไม่สามารถแจงส่วนได้ จะแบ่งการทำงานเป็นออกเป็น 3 ขั้นตอนดังนี้

1. ขั้นตอนการแทนที่สายอักขระของตัวอย่างบวกที่สามารถแปลงเป็นตัวแปรที่มีอยู่ก่อนหน้านี้
2. ขั้นตอนการสร้างกฎใหม่ โดยเปรียบเทียบตัวอย่างบวกที่ถูกแทนที่ด้วยตัวแปร กับสายอักขระของกฎทางขวามือทุกกฎที่กฎทางซ้ายมือเป็นตัวแปรเริ่มต้น S เพื่อหาสายอักขระย่อยที่มีความยาวมากที่สุดที่เกิดขึ้น แล้วนำเอาสายอักขระของตัวอย่างบวกที่พิจารณาอยู่ และสายอักขระที่เปรียบเทียบ มาทำการหาสายอักขระที่เหลือจากการแทนสายอักขระย่อยที่ยาวที่สุด แล้วนำส่วนที่เหลือจากการแทนสายอักขระย่อยมาเปรียบเทียบเพื่อสร้างเป็นกฎวนซ้ำ แล้วจึงนำกฎใหม่ที่สร้างไปรวมกับกฎของไวยากรณ์ที่มีอยู่แล้ว ได้เป็นไวยากรณ์สมมติฐาน จากนั้นนำไวยากรณ์สมมติฐานไปทดสอบการแจงส่วนกับข้อมูลตัวอย่างลบ ถ้าเกิดมีข้อมูลตัวอย่างลบตัวใดที่สามารถแจงส่วนได้จากไวยากรณ์สมมติฐานจะไม่ยอมรับไวยากรณ์สมมติฐาน แต่ถ้าทุก

ตัวอย่างลบไม่สามารถทำการแจงส่วนได้จากไวยากรณ์สมมติฐาน จะกำหนดไวยากรณ์สมมติฐานให้เป็นไวยากรณ์ที่ถูกต้อง

3. ทำการยุบรวมตัวแปรที่สามารถยุบรวมกันได้จากการไวยากรณ์ใหม่เป็นขั้นตอนสุดท้าย และถือเป็นการสิ้นสุดรอบการทำงานจากตัวอย่างบวกเพียง 1 ตัว

ทำการวนตรวจสอบทุกๆข้อมูลนำเข้าจนหมดตัวอย่างบวก จะได้ไวยากรณ์ไม่พึ่งบริบทที่สามารถแจงส่วนได้แต่ตัวอย่างบวกและไม่สามารถแจงส่วนตัวอย่างลบได้เลย จากคำอธิบายข้างต้นสามารถสร้างเป็นรหัสเทียมของอัลกอริทึม ได้ดังนี้

Algorithm

Input: an ordered set S_P of positive sample strings.
an ordered set S_N of negative sample strings.

Output: A set P of rules such that all the strings in S_P are derived from P but no string in S_N is derived from P.

begin

```
for i = 1 to |Sp| do
    if can_not_parse(Sp[i]) with TG then
        Replace_Terminal_with_Nonterminal(Sp[i],TG)
        Create_Rule(Sp[i],TG)
        Merge_Rule(TG)
    endif
endfor
P ← TG
Return P
```

end

ขั้นตอนของการ Replace_Terminal_with_Nonterminal คือ อัลกอริทึมการแทนสายอักขระด้วยตัวแปร Create_Rule คือ อัลกอริทึมการสร้างตัวแปรวนซ้ำโดยวิธีหาสายอักขระย่อยที่มีความยาวมากสุด และ Merge_Rule คือ อัลกอริทึมการผสานตัวแปรใหม่ จะอธิบายการทำงานแต่ละอัลกอริทึมในหัวข้อถัดไป

3.4 อัลกอริทึมการแทนสายอักขระด้วยตัวแปร

เนื่องจากอัลกอริทึมนี้เป็นการแทนสายอักขระของตัวอย่างบวกด้วยตัวแปรของไวยากรณ์ที่มีอยู่ก่อนแล้ว โดยตัวแปรที่นำมาแทนนั้นมีลักษณะวนซ้ำได้ ดังนั้นต้องอาศัยตัวอย่างลบเพื่อมาคัดเลือกว่าตัวแปรใดสามารถแทนที่สายอักขระย่อยใดได้บ้าง ในที่นี้เราได้นำเสนอวิธีการเลือกตัวแปรที่แทนสายอักขระย่อยได้ยาวที่สุด โดยไม่สามารถทำการแจงส่วนตัวอย่างลบได้ทุกตัว ทำการวนซ้ำไปเรื่อยๆจนกว่าจะไม่สามารถแทนสายอักขระได้ด้วยตัวแปรใดๆ โดยการทำงานเริ่มจากจะแบ่งตัวแปรออกเป็นชั้น นิยามชั้นของตัวแปรตามลักษณะของกฎดังนี้

นิยาม 3.1 *กำหนดให้ TG = <V, T, P, S> เป็นไวยากรณ์ไม่พึ่งบริบทที่สร้างขึ้นโดยอัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบทเราสามารถนิยามชั้นของตัวแปรได้ดังนี้*

ถ้า $A \in V, A \neq S$ *โดยที่ทุก* $\alpha, \beta \in T^*$, $A \rightarrow \alpha \in P$ *และ* $A \rightarrow \alpha A\beta \in P$ *แล้ว ให้ A เป็น*ตัวแปรในชั้นที่ 1

ถ้า $B \in V, B \neq S$ *โดยที่ทุก* $\alpha \in (V \cup T)^*$, $B \rightarrow \alpha$ *มี* $C \in V$ *และ C เป็นตัวแปรที่มีค่าชั้นที่มีค่ามากสุด k ในรูปประโยคของ* α *แล้วจะได้ว่า B เป็น*ตัวแปรชั้นที่ k+1

ตัวอย่างที่ 3.1 *กำหนดให้ไวยากรณ์ไม่พึ่งบริบท G = <V, T, P, S> โดยที่*

$V = \{S,A,B,C\}$

$T = \{a,b,c\}$

$P = \{S \rightarrow AB,\ A \rightarrow aaCcc,\ A \rightarrow aaBcc,\ B \rightarrow Cbb,\ C \rightarrow bb\}$

วิธีทำ

จากนิยาม 3.1 จะได้ว่า C จะเป็นตัวแปรชั้นที่ 1 และ B มี C เป็นส่วนหนึ่งของกฎทางขวาจึงเป็นตัวแปรชั้นที่ 2 และ A เป็นซึ่งมี B และ C เป็นส่วนหนึ่งของกฎทางขวาแต่ B เป็นตัวแปรที่มีค่าชั้นมากที่สุดคือ 2 ดังนั้น ตัวแปร A จึงเป็นตัวแปรชั้นที่ 3

□

การค้นหาสายอักขระย่อยที่สามารถแปลงเป็นตัวแปร จะเริ่มจากตัวแปรชั้นที่ 1 ทุก $A \rightarrow \alpha$ โดยที่ $A \in V$, $A \neq S$, $\alpha \in T^+$ ทำการค้นหา α ตัวแรกที่ปรากฏในสายอักขระจากทางซ้าย แล้วแทน α ด้วย A แล้วนำไปตรวจสอบกับตัวอย่างลบ ถ้าไม่มีตัวอย่างลบตัวใดที่สามารถแจงส่วนได้ จะแทน α ด้วย A ถ้ามีตัวอย่างลบตัวใดที่สามารถแจงส่วนได้ จะไม่ทำการแทน α ด้วย A แต่จะทำการค้นหา α ตำแหน่งถัดไป เมื่อทำการแทน α ด้วย A แล้ว ถ้ามีกฎของ A อยู่ในรูปของ $A \rightarrow A\alpha$ จะทำการวนหาอักขระตัวถัดไป ถ้าอักขระตัวถัดไปเป็น α เราจะแทน α ด้วยอักขระว่าง แล้วนำไปทดสอบกับตัวอย่างลบเช่นเดียวกับกรณีแทน α ด้วย A วน

ซ้ำจนกระทั่งไม่เจอ α เป็นตัวถัดไป แต่ถ้ามีกฎอยู่ในรูปของ $A \rightarrow \alpha$, $A \rightarrow \beta A\gamma$ โดยที่ α, β, $\gamma \in T^+$ จะได้ว่า เราจะค้นหา $\beta A\gamma$ ที่ตำแหน่งเดิมที่พบ α แล้วแทน $\beta A\gamma$ ด้วย A แล้วนำไปทดสอบกับตัวอย่างลบ วนซ้ำจนกระทั่งจะไม่พบ $\beta A\gamma$ แทนค่าเช่นนี้จนครบทุกตัวแปร จนกระทั่งไม่สามารถทำการแทนตัวแปรในชั้นที่ 1 ได้ แล้วจึงทำการวนรอบถัดไป ทำเช่นนี้จนครบทุกชั้นของตัวแปร

ตัวอย่างที่ 3.2 *กำหนดให้ไวยากรณ์ไม่พึ่งบริบท G = <V, T, P, S> โดยที่*

V = {S,A,B,C}

T = {a,b,c}

P = {S $\rightarrow$ C, C $\rightarrow$ CAB, C $\rightarrow$ AB, A $\rightarrow$ aAb , A $\rightarrow$ c, B $\rightarrow$ Ba, B $\rightarrow$ a }

S_N *= { aacbbbbbc, aacbbcacb, acbaacbacb } เป็นเซตของตัวอย่างลบ*

S_p *= aaaaacbbbbbacc เป็นสายอักขระของตัวอย่างบวกที่กำลังพิจารณา*

วิธีทำ

จากนิยาม 3.1 จะได้ว่าตัวแปร A,B เป็นสมาชิกของตัวแปรชั้นที่ 1 และตัวแปร C จะเป็นสมาชิกของตัวแปรชั้นที่ 2

รอบที่ 1 ทำการค้นหาการแทนตัวแปรในชั้นที่ 1

เริ่มจากตัวแปรตัวใดก่อนก็ได้ในที่นี้จะเริ่มจากตัวแปร A ดังนั้นต้องทำการหา c ตัวแรก

จะได้ว่า aaaaacbbbbbacc จะสามารถแปลงได้เป็น aaaaaAbbbbbacc

ดังนั้นสร้างไวยากรณ์สมมติฐานได้เป็น {S $\rightarrow$ aaaaaAbbbbbacc } $\cup$ P

เมื่อนำไวยากรณ์ทดสอบไปตรวจสอบกับตัวอย่างลบจะได้ว่า ไม่สามารถทำการแจงส่วนไปยังตัวอย่างลบได้ทุกตัวดังนั้นเราจึงเปลี่ยนสายอักขระให้เป็น

S_p = aaaaaAbbbbbacc

เมื่อตัวแปร A อยู่ในรูปแบบของ $A \rightarrow c$, $A \rightarrow aAb$ จะได้ว่าเราจะทำหาตัวแปร aAb ที่ตำแหน่งเดิม เปลี่ยนสายอักขระเป็น aaaaAbbbbacc

ดังนั้นสร้างไวยากรณ์สมมติฐานได้เป็น {S $\rightarrow$ aaaaAbbbbacc } $\cup$ P

เมื่อนำไวยากรณ์ทดสอบไปตรวจสอบกับตัวอย่างลบจะได้ว่า ไม่สามารถทำการแจงส่วนไปยังตัวอย่างลบได้ทุกตัวดังนั้นเราจึงเปลี่ยนสายอักขระให้เป็น

S_p = aaaaAbbbbacc

วนทำไปเรื่อยๆจะได้สายอักขระสุดท้ายของการแปลงสายอักขระเป็น A คือ

S_p = Aacc

ทำการค้นหา c ตัวถัดไป

จะได้ว่า Aacc จะสามารถแปลงได้เป็น AaAc

ดังนั้นสร้างไวยากรณ์สมมติฐานได้เป็น {S $\longrightarrow$ AaAc } $\cup$ P

เมื่อนำไวยากรณ์ทดสอบไปตรวจสอบกับตัวอย่างลบจะได้ว่า ไม่สามารถทำการแจงส่วนไปยังตัวอย่างลบได้ทุกตัวดังนั้นเราจึงเปลี่ยนสายอักขระให้เป็น

S_p = AaAc

เมื่อ A ตัวแปร A อยู่ในรูปแบบของ A $\longrightarrow$ c, A $\longrightarrow$ aAb จะได้ว่าเราจะทำหาตัวแปร aAb ที่ตำแหน่งเดิม แต่ไม่พบดังนั้น

ทำการค้นหา c ตัวถัดไป

จะได้ว่า AaAc จะสามารถแปลงได้เป็น AaAA

ดังนั้นสร้างไวยากรณ์สมมติฐานได้เป็น {S $\longrightarrow$ AaAA } $\cup$ P

เมื่อนำไวยากรณ์ทดสอบไปตรวจสอบกับตัวอย่างลบจะได้ว่า มีตัวอย่างลบสามารถทำการแจงส่วนได้ คือ acbaacbacb ดังนั้นเราจึงไม่ทำการเปลี่ยนสายอักขระให้เป็น AaAA ดังนั้น S_p ยังคงเป็น AaAc

ทำการค้นหา c ที่ตำแหน่งถัดไปพบว่าไม่มีแล้ว

เปลี่ยนตัวแปรที่พิจารณาเป็น B ทำการค้นหา a ตัวแรก

จะได้ว่า AaAc สามารถเปลี่ยนได้เป็น ABAc

ดังนั้นสร้างไวยากรณ์สมมติฐานได้เป็น {S $\longrightarrow$ ABAc } $\cup$ P

เมื่อนำไวยากรณ์ทดสอบไปตรวจสอบกับตัวอย่างลบจะได้ว่า ไม่สามารถทำการแจงส่วนไปยังตัวอย่างลบได้ทุกตัวดังนั้นเราจึงเปลี่ยนสายอักขระให้เป็น

S_p = ABAc

ทำการค้นหา a ที่ตำแหน่งถัดไปพบว่าไม่มีแล้ว และไม่มีตัวแปรในชั้นที่ 1 แล้วจึงสิ้นสุดรอบที่ 1

รอบที่ 2 ทำการค้นหาการแทนตัวแปรในชั้นที่ 2

เริ่มจากตัวแปร C หา AB ตัวแรก

จะได้ว่า ABAc จะสามารถแปลงได้เป็น CAc

ดังนั้นสร้างไวยากรณ์สมมติฐานได้เป็น {S $\longrightarrow$ CAc } $\cup$ P

เมื่อนำไวยากรณ์ทดสอบไปตรวจสอบกับตัวอย่างลบจะได้ว่า ไม่สามารถทำการแจงส่วนไปยังตัวอย่างลบได้ทุกตัวดังนั้นเราจึงเปลี่ยนสายอักขระให้เป็น

S_p = CAc

ทำการค้นหา AB ที่ตำแหน่งถัดไปพบว่าไม่มีแล้วและไม่มีตัวแปรอื่นที่ไม่ใช่ตัวแปรเริ่มต้นอีกแล้ว

ดังนั้นสุดท้ายจากอัลกอริทึมการแปลงสายอักขระด้วยตัวแปรจะได้ว่า เราสามารถแปลงสายอักขระให้เป็นตัวแปรได้เป็น

S_p = CAc □

3.5 อัลกอริทึมการสร้างตัวแปรวนซ้ำโดยวิธีหาสายอักขระย่อยที่มีความยาวมากสุด

เรานำสายอักขระตัวอย่างบวกที่ผ่านการแทนสายอักขระด้วยตัวแปร แทนด้วย S_{long} มาหาสายอักขระย่อยที่ยาวที่สุดเมื่อเทียบกับแต่ละสายอักขระทางขวามือของกฎที่มีกฎทางซ้ายมือเป็นตัวแปรเริ่มต้น ให้กฎแทนด้วย $S \rightarrow S_{short}$ จะได้ว่าสายอักขระทางขวามือของกฎที่มีกฎทางซ้ายมือเป็นตัวแปรเริ่มต้น แทนด้วย S_{short} ซึ่งแต่ละคู่ของ S_{short} และ S_{long} สามารถหาสายอักขระย่อยยาวสุดได้ โดยหาเซตสายอักขระย่อยของ S_{short} ทั้งหมดที่เป็นไปได้ นำสมาชิกแต่ละตัวไปค้นหา ว่าเป็นสายอักขระย่อยใน S_{long} หรือไม่โดยใช้อัลกอริทึมของคนุท มอริส แพรต (Knuth-Morris-Pratt algorithm)[17] หลังจากทำการเปรียบเทียบสายอักขระย่อยที่ยาวที่สุดแทนด้วย β จะเรียงลำดับการพิจารณาสายอักขระ S_{short}, S_{long} จากค่า $|\beta|$ ที่มากที่สุดก่อน ซึ่งกับสายอักขระ S_{short}, S_{long} เราจะแบ่งแต่ละสายอักขระออกได้เป็นสามส่วนคือ

$$S_{short} = \alpha_1\beta\gamma_1$$

$$S_{long} = \alpha_2\beta\gamma_2$$

ให้ α_1, γ_1 , α_2, γ_2 เป็นสายอักขระที่เหลือจากการแทน S_{short}, S_{long} ด้วยสายอักขระย่อย β ตามลำดับ

เมื่อแบ่งสายอักขระออกเป็น 3 ส่วนแล้วนำ α_1, α_2 และ γ_1, γ_2 มาพิจารณาตัวแปรวนซ้ำทีละส่วนซึ่งประกอบด้วยการทำงาน 5 ขั้นตอนคือ

ขั้นตอนที่ 1 การเปรียบเทียบระหว่าง α_1, α_2 ซึ่งจะสามารถพิจารณาได้ออกเป็น 3 กรณีดังนี้

กรณีที่ 1 ถ้า $\alpha_1 \neq \lambda$, $\alpha_2 \neq \lambda$

เราจะทำการสร้างกฎวนซ้ำที่มีรูปแบบดังนี้

$$S \rightarrow A\beta\gamma_1$$

$$S \rightarrow A\beta\gamma_2$$

$$A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \alpha_1 \mid \alpha_2$$

โดยที่ ตัวแปร A เป็นตัวแปรใหม่ที่ไม่ซ้ำกับตัวแปรที่มีอยู่ในไวยากรณ์

จากนั้นนำกฎวนซ้ำที่ได้ไปรวมกับกฎที่มีอยู่ก่อนเป็นไวยากรณ์สมมติฐาน แล้วนำไวยากรณ์สมมติที่ได้ไปทดสอบการแจงส่วนตัวอย่างลบทุกตัว ผลที่ได้จากการทดสอบการแจงส่วน จะบอกถึงไวยากรณ์สมมติฐานว่าที่เป็นไวยากรณ์ที่ถูกต้อง ก็ต่อเมื่อไม่สามารถทำการแจงส่วนกับตัวอย่างลบทุกตัวได้ ดังนั้นเราจะทำการเปลี่ยนกฎของไวยากรณ์ที่ถูกต้องเป็น

$$\{S \rightarrow A\beta\gamma_1,\ S \rightarrow A\beta\gamma_2,\ A \rightarrow A\alpha_1,\ A \rightarrow A\alpha_2,\ A \rightarrow \alpha_2,\ A \rightarrow \alpha_1\ \} \cup TG - \{\ S \rightarrow S_{short}\ \}$$

เมื่อทำการเปลี่ยนกฎเรียบร้อยจึงไปสู่ขั้นตอนที่ 2 ต่อไป แต่ถ้าผลการทดสอบตัวอย่างลบ สามารถทำการแจงส่วนตัวอย่างลบได้แม้เพียงตัวเดียว จะทำการพิจารณา α_1, α_2 ดังนี้

ถ้า α_1, α_2 อยู่ในรูปของ $\alpha_1 = \alpha_3\alpha_4$, $\alpha_2 = \alpha_3 v \alpha_4 x$

โดยที่ α_3, $\alpha_4 \in (V \cup T)^*$ และ $v,x \in T^+$

แล้วเราจะสร้างกฎวนซ้ำได้เป็น

$S \longrightarrow \alpha_3 A \beta \gamma_1$

$S \longrightarrow \alpha_3 A \beta \gamma_2$

$A \longrightarrow vAx \mid \alpha_4$

โดยที่ ตัวแปร A เป็นตัวแปรใหม่ที่ไม่ซ้ำกับตัวแปรที่มีอยู่ในไวยากรณ์

จากนั้นนำกฎวนซ้ำที่ได้ไปรวมกับกฎที่มีอยู่ก่อนเป็นไวยากรณ์สมมติฐาน แล้วนำไวยากรณ์สมมติที่ได้ไปทดสอบการแจงส่วนตัวอย่างลบทุกตัว ผลที่ได้จากการทดสอบการแจงส่วน จะบอกถึงไวยากรณ์สมมติฐานว่าที่เป็นไวยากรณ์ที่ถูกต้อง ก็ต่อเมื่อไม่สามารถทำการแจงส่วนกับตัวอย่างลบทุกตัวได้ ดังนั้นเราจะทำการเปลี่ยนกฎของไวยากรณ์ที่ถูกต้องเป็น

$\{S \longrightarrow \alpha_3 A \beta \gamma_1,\ S \longrightarrow \alpha_3 A \beta \gamma_2,\ A \longrightarrow vAx,\ A \longrightarrow \alpha_4\ \} \cup TG - \{$

$S \longrightarrow S_{short}\ \}$

เมื่อทำการเปลี่ยนกฎเรียบร้อยจึงไปสู่ขั้นตอนที่ 2 ต่อไป แต่ถ้าผลการทดสอบตัวอย่างลบ สามารถทำการแจงส่วนตัวอย่างลบได้แม้เพียงตัวเดียว จะไปพิจารณากรณีที่ 2 แทน

กรณีที่ 2 จะทำการพิจารณา ก็ต่อเมื่อ ถ้าไม่เข้ากรณีแรกหรือเข้ากรณีแรกแต่ไม่สามารถสร้างกฎเพิ่มได้จากกรณีแรก

พิจารณา α_1, α_2 ถ้า α_1, α_2 อยู่ในรูปของ $\alpha_1 = \alpha_3$, $\alpha_2 = \alpha_3 v$

โดยที่ $\alpha_3 \in (V \cup T)^*$ และ $v \in T^+$

เราจะพิจารณาการวนซ้ำสร้างกฎทั้งหมด 3 แบบดังนี้

แบบที่ 1 ถ้า $|v| > 1$ เราจะวนสร้างกฎวนซ้ำดังนี้

$S \longrightarrow \alpha_3 A \beta \gamma_1$

$S \longrightarrow \alpha_3 A \beta \gamma_2$

$A \longrightarrow Av \mid \lambda$

$A \longrightarrow v_1 \ldots v_i A v_{i+1} \ldots v_n$ ทุก $i < n$, $n = |v|$, $v_j \in T$

โดยที่ $v = v_1 \ldots v_i v_{i+1} \ldots v_n$

จากนั้นนำกฎวนซ้ำที่ได้ไปรวมกับกฎที่มีอยู่ก่อนเป็นไวยากรณ์สมมติฐาน แล้วนำไวยากรณ์สมมติที่ได้ไปทดสอบการแจงส่วนตัวอย่างลบทุกตัว ผลที่ได้จากการทดสอบการแจงส่วน จะบอกถึงไวยากรณ์สมมติฐานว่าที่เป็นไวยากรณ์ที่ถูกต้อง ก็ต่อเมื่อไม่สามารถทำการแจงส่วนกับตัวอย่างลบทุกตัวได้ ดังนั้นเราจะทำการเปลี่ยนกฎของไวยากรณ์ที่ถูกต้องเป็น

$\{S \longrightarrow uA\beta\gamma_1,\ S \longrightarrow uA\beta\gamma_2,\ A \longrightarrow Av,\ A \longrightarrow v_1...v_iAv_{i+1}...v_n,\ A \longrightarrow \lambda\ \} \cup TG - \{ S \longrightarrow S_{short} \}$

เมื่อทำการเปลี่ยนกฎเรียบร้อยจึงไปสู่ขั้นตอนที่ 2 ต่อไป แต่ถ้ามีตัวอย่างลบแม้เพียงตัวเดียวที่สามารถทำการแจงส่วนด้วยไวยากรณ์สมมติฐานจะวนทำจนกว่าหา i ที่สร้างไวยากรณ์สมมติฐานได้ถูกต้อง ถ้าไม่มี i ที่น้อยกว่า n ที่สร้างไวยากรณ์สมมติฐานได้ถูกต้อง เราจะพิจารณาวนสร้างกฎวนซ้ำแบบที่ 2

แบบที่ 2 ถ้า $|v| > 1$ เราจะวนสร้างกฎวนซ้ำอีกครั้งดังนี้

$S \longrightarrow \alpha_3A\beta\gamma_1$

$S \longrightarrow \alpha_3A\beta\gamma_2$

$A \longrightarrow v_1...v_iAv_{i+1}...v_n \mid \lambda$ ทุก $i < n$, $n = |v|$, $v_j \in T$

โดยที่ $v = v_1...v_iv_{i+1}...v_n$

จากนั้นนำกฎวนซ้ำที่ได้ไปรวมกับกฎที่มีอยู่ก่อนเป็นไวยากรณ์สมมติฐาน แล้วนำไวยากรณ์สมมติที่ได้ไปทดสอบการแจงส่วนตัวอย่างลบทุกตัว ผลที่ได้จากการทดสอบการแจงส่วน จะบอกถึงไวยากรณ์สมมติฐานว่าที่เป็นไวยากรณ์ที่ถูกต้อง ก็ต่อเมื่อไม่สามารถทำการแจงส่วนกับตัวอย่างลบทุกตัวได้ ดังนั้นเราจะทำการเปลี่ยนกฎของไวยากรณ์ที่ถูกต้องเป็น

$\{S \longrightarrow uA\beta\gamma_1,\ S \longrightarrow uA\beta\gamma_2,\ A \longrightarrow v_1...v_iAv_{i+1}...v_n,\ A \longrightarrow \lambda\ \} \cup TG - \{ S \longrightarrow S_{short} \}$

เมื่อทำการเปลี่ยนกฎเรียบร้อยจึงไปสู่ขั้นตอนที่ 2 ต่อไป แต่ถ้ามีตัวอย่างลบแม้เพียงตัวเดียวที่สามารถทำการแจงส่วนด้วยไวยากรณ์สมมติฐานจะวนทำจนกว่าหา i ที่สร้างไวยากรณ์สมมติฐานได้ถูกต้อง ถ้าไม่มี i ที่น้อยกว่า n ที่สร้างไวยากรณ์สมมติฐานได้ถูกต้อง เราจะพิจารณาสร้างกฎวนซ้ำแบบที่ 3

แบบที่ 3 เราจะสร้างกฎดังนี้

$S \rightarrow \alpha_3 A \beta \gamma_1$

$S \rightarrow \alpha_3 A \beta \gamma_2$

$A \rightarrow Av \mid \lambda$

จากนั้นนำกฎวนซ้ำที่ได้ไปรวมกับกฎที่มีอยู่ก่อนเป็นไวยากรณ์สมมติฐาน แล้วนำไวยากรณ์สมมติที่ได้ไปทดสอบการแจงส่วนตัวอย่างลบทุกตัว ผลที่ได้จากการทดสอบการแจงส่วน จะบอกถึงไวยากรณ์สมมติฐานว่าที่เป็นไวยากรณ์ที่ถูกต้อง ก็ต่อเมื่อไม่สามารถทำการแจงส่วนกับตัวอย่างลบทุกตัวได้ ดังนั้นเราจะทำการเปลี่ยนกฎของไวยากรณ์ที่ถูกต้องเป็น

$\{S \rightarrow uA\beta\gamma_1,\ S \rightarrow uA\beta\gamma_2,\ A \rightarrow Av,\ A \rightarrow \lambda\ \} \cup TG - \{ S \rightarrow S_{short} \}$

เมื่อทำการเปลี่ยนกฎเรียบร้อยจึงไปสู่ขั้นตอนที่ 2 ต่อไป แต่ถ้ามีตัวอย่างลบแม้เพียงตัวเดียวที่สามารถทำการแจงส่วนด้วยไวยากรณ์สมมติฐานจะพิจารณากรณีที่ 3

กรณีที่ 3 เมื่อไม่เข้า 2 กรณีแรก หรือไม่สามารถสร้างกฎใหม่ได้จาก 2 กรณีแรก จะไม่ทำการเปลี่ยนแปลงใดๆ และทำการพิจารณา γ_1, γ_2 ในขั้นตอนที่ 2 ถัดไป

ตัวอย่าง 3.3 *การพิจารณา* a_1, a_2

กำหนดให้ $S_{short} = bbcc$

$S_{long} = bbacc$

วิธีทำ

จากการเปรียบเทียบระหว่าง S_{short}, S_{long} จะได้ว่า

$\alpha_1 = bb,\ \alpha_2 = bba,\ \gamma_1 = \lambda,\ \gamma_2 = \lambda,\ \beta = cc$

เมื่อ $\alpha_1 \neq \lambda$, $\alpha_2 \neq \lambda$ ดังนั้นจะเข้าสู่กรณีแรก

สามารถสร้างกฎวนซ้ำได้ดังนี้

$S \rightarrow Acc$

$A \rightarrow Abba \mid Abb \mid bb \mid bba$

จากนั้นนำกฎวนซ้ำที่ได้ไปสร้างเป็นไวยากรณ์สมมติฐานแล้วนำไปทดสอบกับตัวอย่างลบทั้งหมด ถ้าไม่มีตัวอย่างลบตัวใดที่สามารถแจงส่วนได้จะทำการเปลี่ยนกฎเป็น

$\{ S \rightarrow Acc,\ A \rightarrow Abba,\ A \rightarrow Abb,\ A \rightarrow bba,\ A \rightarrow bb \} \cup TG - \{ S \rightarrow bbacc \}$

เมื่อทำการสร้างกฎใหม่ได้แล้วจึงจบการทำงานของขั้นตอนที่ 1

ในกรณีที่มีตัวอย่างลบแม้เพียงหนึ่งตัวที่สามารถแจงส่วนได้ จะพิจารณา α_1, α_2 ว่าอยู่ในรูปของ $\alpha_1 = \alpha_3\alpha_4$, $\alpha_2 = \alpha_3 v \alpha_4 x$ โดยที่ α_3, $\alpha_4 \in (V \cup T)^*$ และ $v,x \in T^+$ หรือไม่ ซึ่งไม่อยู่ในรูปของ $\alpha_1 = \alpha_3\alpha_4$, $\alpha_2 = \alpha_3 v \alpha_4 x$ ดังนั้นจึงไม่เข้ากรณีแรก

ถ้าในกรณีนี้ กรณีแรกไม่สามารถสร้างกฎใหม่ได้

ดังนั้นจึงเข้าสู่กรณีที่สอง

พิจารณา α_1, α_2 อยู่ในรูปของ $\alpha_1 = \alpha_3$, $\alpha_2 = \alpha_3 v$ โดยที่ $\alpha_3 \in (V \cup T)^*$ และ $v \in T^+$ จะได้ว่าเราจะสร้างกฎวนซ้ำคือ

$S \longrightarrow bbAcc$

$A \longrightarrow Aa \mid \lambda$

จากนั้นนำกฎวนซ้ำที่ได้ไปสร้างเป็นไวยากรณ์สมมติฐานแล้วนำไปทดสอบกับตัวอย่างลบทั้งหมด ถ้าไม่มีตัวอย่างลบตัวใดที่สามารถแจงส่วนได้จะทำการเปลี่ยนกฎเป็น

$\{ S \longrightarrow bbAcc, A \longrightarrow Aa, A \longrightarrow \lambda \} \cup TG - \{ S \longrightarrow bbacc \}$

เมื่อทำการสร้างกฎใหม่ได้แล้วจึงจบการทำงานของขั้นตอนที่ 1

ในกรณีที่มีตัวอย่างลบตัวแม้เพียงหนึ่งตัวที่สามารถแจงส่วนได้จะไม่ทำการเปลี่ยนแปลง

□

ขั้นตอนที่ 2 การเปรียบเทียบระหว่าง γ_1, γ_2 ซึ่งจะสามารถพิจารณาได้ออกเป็น 3 กรณีดังนี้

กรณีที่ 1 ถ้า $\gamma_1 \neq \lambda$, $\gamma_2 \neq \lambda$

เราจะทำการสร้างกฎวนซ้ำที่มีรูปแบบดังนี้

$S \longrightarrow \alpha_1 \beta C$

$S \longrightarrow \alpha_2 \beta C$

$C \longrightarrow C\gamma_1 \mid C\gamma_2 \mid \gamma_1 \mid \gamma_2$

แต่ถ้ามีการเพิ่มกฎวนซ้ำจากขั้นตอนที่ 1 จะสร้างกฎวนซ้ำได้เป็นรูปแบบนี้แทน

$S \longrightarrow A \beta C$

$C \longrightarrow C\gamma_1 \mid C\gamma_2 \mid \gamma_1 \mid \gamma_2$

โดยที่ A เป็นส่วนที่เพิ่มกฎวนซ้ำจากขั้นตอนที่ 1 และตัวแปร C เป็นตัวแปรใหม่ที่ไม่ซ้ำกับตัวแปรที่มีอยู่ในไวยากรณ์

จากนั้นนำกฎวนซ้ำที่ได้ไปรวมกับกฎที่มีอยู่ก่อนเป็นไวยากรณ์สมมติฐาน แล้วนำไวยากรณ์สมมติที่ได้ไปทดสอบการแจงส่วนตัวอย่างลบทุกตัว ผลที่ได้จากการทดสอบการแจงส่วน จะบอกถึงไวยากรณ์สมมติฐานว่าที่เป็นไวยากรณ์ที่ถูกต้อง ก็ต่อเมื่อไม่สามารถทำการแจงส่วนกับตัวอย่างลบทุกตัวได้ ดังนั้นเราจะทำการเปลี่ยนกฎของไวยากรณ์ที่ถูกต้องเป็น

$\{S \rightarrow \alpha_1\beta C,\ S \rightarrow \alpha_2\beta C,\ C \rightarrow C\gamma_1,\ C \rightarrow C\gamma_2,\ C \rightarrow \gamma_1,\ C \rightarrow \gamma_2\} \cup TG - \{S \rightarrow S_{short}\}$

แต่ถ้ามีการเพิ่มกฎวนซ้ำจากขั้นตอนที่ 1 จะเปลี่ยนกฎด้วย

$\{S \rightarrow A\beta C,\ C \rightarrow C\gamma_1,\ C \rightarrow C\gamma_2,\ C \rightarrow \gamma_2,\ C \rightarrow \lambda\} \cup TG - \{S \rightarrow S_{short}\}$

แต่ถ้าผลการทดสอบตัวอย่างลบ สามารถทำการแจงส่วนตัวอย่างลบได้แม้เพียงตัวเดียว จะทำการพิจารณา γ_1, γ_2 ดังนี้

ถ้า $\gamma_1 = \gamma_3\gamma_4$, $\gamma_2 = u\gamma_3 w\gamma_4$

โดยที่ $\gamma_3, \gamma_4 \in (V \cup T)^*$ และ $u,w \in T^+$

แล้วเราจะสร้างกฎวนซ้ำได้เป็น

$S \rightarrow \alpha_1\beta B\gamma_4$

$S \rightarrow \alpha_2\beta B\gamma_4$

$B \rightarrow uBw \mid \gamma_3$

แต่ถ้ามีการเพิ่มกฎวนซ้ำจากขั้นตอนที่ 1 จะสร้างกฎวนซ้ำได้เป็นรูปแบบนี้แทน

$S \rightarrow A\beta B\gamma_4$

$B \rightarrow uBw \mid \gamma_3$

โดยที่ A เป็นส่วนที่เพิ่มกฎวนซ้ำจากขั้นตอนที่ 1 ตัวแปร B เป็นตัวแปรใหม่ที่ไม่ซ้ำกับตัวแปรที่มีอยู่ในไวยากรณ์

จากนั้นนำกฎวนซ้ำที่ได้ไปรวมกับกฎที่มีอยู่ก่อนเป็นไวยากรณ์สมมติฐาน แล้วนำไวยากรณ์สมมติที่ได้ไปทดสอบการแจงส่วนตัวอย่างลบทุกตัว ผลที่ได้จากการทดสอบการแจงส่วน จะบอกถึงไวยากรณ์สมมติฐานว่าที่เป็นไวยากรณ์ที่ถูกต้อง ก็ต่อเมื่อไม่สามารถทำการแจงส่วนกับตัวอย่างลบทุกตัวได้ ดังนั้นเราจะทำการเปลี่ยนกฎของไวยากรณ์ที่ถูกต้องเป็น

$\{S \rightarrow \alpha_1\beta B\gamma_4,\ S \rightarrow \alpha_2\beta B\gamma_4,\ B \rightarrow uBw,\ B \rightarrow \gamma_3\ \} \cup TG - \{ S \rightarrow S_{short}\ \}$

แต่ถ้ามีการเพิ่มกฎวนซ้ำจากขั้นตอนที่ 1 จะเปลี่ยนกฎด้วย

$\{S \rightarrow A\beta B\gamma_4,\ B \rightarrow uBw \mid\ B \rightarrow \gamma_3\ \} \cup TG - \{ S \rightarrow S_{short}\ \}$

ถ้ามีตัวอย่างลบแม้เพียงตัวเดียวที่สามารถทำการแจงส่วนด้วยไวยากรณ์สมมติฐาน จะไปพิจารณากรณีที่ 2 แทน

กรณีที่ 2 จะทำการพิจารณา ก็ต่อเมื่อ ถ้าไม่เข้ากรณีแรกหรือเข้ากรณีแรกแต่ไม่สามารถสร้างกฎเพิ่มได้จากกรณีแรก

พิจารณา γ_1, γ_2 ถ้า γ_1, γ_2 อยู่ในรูปของ $\gamma_1 = u\gamma_3,\ \gamma_2 = \gamma_3$

โดยที่ $\gamma_3 \in (V \cup T)^*$ และ $u \in T^+$

เราจะพิจารณาการวนซ้ำสร้างกฎทั้งหมด 3 แบบดังนี้

แบบที่ 1 ถ้า $|u| > 1$ เราจะวนสร้างกฎวนซ้ำดังนี้

$S \rightarrow \alpha_1\beta B\gamma_3$

$S \rightarrow \alpha_2\beta B\gamma_3$

$B \rightarrow Bu \mid\ \lambda$

$B \rightarrow u_1 \ldots u_i B u_{i+1} \ldots u_n$ ทุก $i < n,\ n = |u|,\ u_j \in T$

โดยที่ $u = u_1 \ldots u_i u_{i+1} \ldots u_n$

แต่ถ้ามีการเพิ่มกฎวนซ้ำจากขั้นตอนที่ 1 จะสร้างกฎวนซ้ำได้เป็นรูปแบบนี้แทน

$S \rightarrow A\beta B\gamma_3$

$B \rightarrow Bu \mid\ \lambda$

$B \rightarrow u_1 \ldots u_i B u_{i+1} \ldots u_n$ ทุก $i < n,\ n = |u|,\ u_j \in T$

โดยที่ $u = u_1 \ldots u_i u_{i+1} \ldots u_n$

โดยที่ A เป็นส่วนที่เพิ่มกฎวนซ้ำจากขั้นตอนที่ 1 ตัวแปร B เป็นตัวแปรใหม่ที่ไม่ซ้ำกับตัวแปรที่มีอยู่ในไวยากรณ์

จากนั้นนำกฎวนซ้ำที่ได้ไปรวมกับกฎที่มีอยู่ก่อนเป็นไวยากรณ์สมมติฐาน แล้วนำไวยากรณ์สมมติที่ได้ไปทดสอบการแจงส่วนตัวอย่างลบทุกตัว ผลที่ได้จากการทดสอบการแจงส่วน จะบอกถึงไวยากรณ์สมมติฐานว่าที่เป็นไวยากรณ์ที่ถูกต้อง ก็ต่อเมื่อไม่สามารถทำการแจงส่วนกับตัวอย่างลบทุกตัวได้ ดังนั้นเราจะทำการเปลี่ยนกฎของไวยากรณ์ที่ถูกต้องเป็น

$\{S \longrightarrow \alpha_1\beta B\gamma_3,\ S \longrightarrow \alpha_2\beta B\gamma_3,\ B \longrightarrow uB,\ B \longrightarrow u_1...u_iBu_{i+1}...u_n,$
$B \longrightarrow \lambda\ \} \cup TG - \{\ S \longrightarrow S_{short}\ \}$

แต่ถ้ามีการเพิ่มกฎวนซ้ำจากขั้นตอนที่ 1 จะเปลี่ยนกฎด้วย

$\{S \longrightarrow A\beta B\gamma_3,\ B \longrightarrow uB,\ B \longrightarrow u_1...u_iBu_{i+1}...u_n,\ B \longrightarrow \lambda\ \} \cup TG$
$- \{\ S \longrightarrow S_{short}\ \}$

แต่ถ้ามีตัวอย่างลบแม้เพียงตัวเดียวที่สามารถทำการแจงส่วนด้วยไวยากรณ์สมมติฐาน จะวนทำจนกว่าหา i ที่สร้างไวยากรณ์สมมติฐานได้ถูกต้อง ถ้าไม่มี i ที่น้อยกว่า n ที่สร้างไวยากรณ์สมมติฐานได้ถูกต้อง เราจะพิจารณาวนสร้างกฎวนซ้ำแบบที่ 2

แบบที่ 2 ถ้า $|u| > 1$ เราจะวนสร้างกฎวนซ้ำอีกครั้งดังนี้

$S \longrightarrow \alpha_1\beta B\gamma_3$

$S \longrightarrow \alpha_2\beta B\gamma_3$

$B \longrightarrow u_1...u_iBu_{i+1}...u_n \mid \lambda$ ทุก $i < n$, $n = |u|$, $u_j \in T$

โดยที่ $u = u_1...u_iu_{i+1}...u_n$

แต่ถ้ามีการเพิ่มกฎวนซ้ำจากขั้นตอนที่ 1 จะสร้างกฎวนซ้ำได้เป็นรูปแบบนี้แทน

$S \longrightarrow A\beta B\gamma_3$

$B \longrightarrow u_1...u_iBu_{i+1}...u_n \mid \lambda$ ทุก $i < n$, $n = |u|$, $u_j \in T$

โดยที่ $u = u_1...u_iu_{i+1}...u_n$

โดยที่ A เป็นส่วนที่เพิ่มกฎวนซ้ำจากขั้นตอนที่ 1 ตัวแปร B เป็นตัวแปรใหม่ที่ไม่ซ้ำกับตัวแปรที่มีอยู่ในไวยากรณ์

จากนั้นนำกฎวนซ้ำที่ได้ไปรวมกับกฎที่มีอยู่ก่อนเป็นไวยากรณ์สมมติฐาน แล้วนำไวยากรณ์สมมติที่ได้ไปทดสอบการแจงส่วนตัวอย่างลบทุก

ตัว ผลที่ได้จากการทดสอบการแจงส่วน จะบอกถึงไวยากรณ์สมมติฐานว่าที่เป็นไวยากรณ์ที่ถูกต้อง ก็ต่อเมื่อไม่สามารถทำการแจงส่วนกับตัวอย่างลบทุกตัวได้ ดังนั้นเราจะทำการเปลี่ยนกฎของไวยากรณ์ที่ถูกต้องเป็น

$\{S \rightarrow \alpha_1\beta B\gamma_3,\ S \rightarrow \alpha_2\beta B\gamma_3,\ B \rightarrow u_1...u_iBu_{i+1}...u_n,\ B \rightarrow \lambda\ \} \cup TG - \{\ S \rightarrow S_{short}\ \}$

แต่ถ้ามีการเพิ่มกฎวนซ้ำจากขั้นตอนที่ 1 จะเปลี่ยนกฎด้วย

$\{S \rightarrow A\beta B\gamma_3,\ B \rightarrow u_1...u_iBu_{i+1}...u_n,\ B \rightarrow \lambda\ \} \cup TG - \{\ S \rightarrow S_{short}\ \}$

แต่ถ้ามีตัวอย่างลบแม้เพียงตัวเดียวที่สามารถทำการแจงส่วนด้วยไวยากรณ์สมมติฐาน จะวนทำจนกว่าหา i ที่สร้างไวยากรณ์สมมติฐานได้ถูกต้อง ถ้าไม่มี i ที่น้อยกว่า n ที่สร้างไวยากรณ์สมมติฐานได้ถูกต้อง เราจะพิจารณาวนสร้างกฎวนซ้ำแบบที่ 3

แบบที่ 3 เราจะสร้างกฎวนซ้ำได้เป็น

$S \rightarrow \alpha_1\beta B\gamma_3$

$S \rightarrow \alpha_2\beta B\gamma_3$

$B \rightarrow Bu \mid \lambda$

แต่ถ้ามีการเพิ่มกฎวนซ้ำจากขั้นตอนที่ 1 จะสร้างกฎวนซ้ำได้เป็นรูปแบบนี้แทน

$S \rightarrow A\beta B\gamma_3$

$B \rightarrow uB \mid \lambda$

โดยที่ A เป็นส่วนที่เพิ่มกฎวนซ้ำจากขั้นตอนที่ 1 ตัวแปร B เป็นตัวแปรใหม่ที่ไม่ซ้ำกับตัวแปรที่มีอยู่ในไวยากรณ์

จากนั้นนำกฎวนซ้ำที่ได้ไปรวมกับกฎที่มีอยู่ก่อนเป็นไวยากรณ์สมมติฐาน แล้วนำไวยากรณ์สมมติที่ได้ไปทดสอบการแจงส่วนตัวอย่างลบทุกตัว ผลที่ได้จากการทดสอบการแจงส่วน จะบอกถึงไวยากรณ์สมมติฐานว่าที่เป็นไวยากรณ์ที่ถูกต้อง ก็ต่อเมื่อไม่สามารถทำการแจงส่วนกับตัวอย่างลบทุกตัวได้ ดังนั้นเราจะทำการเปลี่ยนกฎของไวยากรณ์ที่ถูกต้องเป็น

$\{S \rightarrow \alpha_1\beta B\gamma_3,\ S \rightarrow \alpha_2\beta B\gamma_3,\ B \rightarrow uB,\ B \rightarrow \lambda\ \} \cup TG - \{\ S \rightarrow S_{short}\ \}$

แต่ถ้ามีการเพิ่มกฎวนซ้ำจากขั้นตอนที่ 1 จะเพิ่มกฎเป็น

$\{S \rightarrow A\beta B\gamma_3, B \rightarrow uB, B \rightarrow \lambda\} \cup TG - \{S \rightarrow S_{short}\}$

ถ้ามีตัวอย่างลบแม้เพียงตัวเดียวที่สามารถทำการแจงส่วนด้วยไวยากรณ์สมมติฐาน จะไปไม่ทำการเปลี่ยนแปลงใดๆ

กรณีที่ 3 เมื่อไม่เข้า 2 กรณีแรก จะไม่ทำการเปลี่ยนแปลงใดๆ

ตัวอย่าง 3.4 *การพิจารณา* γ_1, γ_2

กำหนดให้ $S_{short} = ccbb$

$S_{long} = ccabbb$

วิธีทำ

จากการเปรียบเทียบระหว่าง S_{short}, S_{long} จะได้ว่า

$\alpha_1 = \lambda, \alpha_2 = \lambda, \gamma_1 = bb, \gamma_2 = abbb, \beta = cc$

เมื่อ $\gamma_1 \neq \lambda$, $\gamma_2 \neq \lambda$ ดังนั้นจะเข้าสู่กรณีแรก

จะได้ว่าเราจะสร้างกฎวนซ้ำคือ

$S \rightarrow ccA$

$A \rightarrow Aabbb \mid Abb \mid bb \mid abbb$

จากนั้นนำกฎวนซ้ำที่ได้ไปสร้างเป็นไวยากรณ์สมมติฐานแล้วนำไปทดสอบกับตัวอย่างลบทั้งหมด ถ้าไม่มีตัวอย่างลบตัวใดที่สามารถแจงส่วนได้จะทำการเปลี่ยนกฎเป็น

$\{S \rightarrow ccA, A \rightarrow Aabbb, A \rightarrow Abb, A \rightarrow abbb, A \rightarrow bb\} \cup TG - \{$

$S \rightarrow ccabbb\}$

เมื่อทำการสร้างกฎใหม่ได้แล้วจึงจบการทำงานของขั้นตอนที่ 2

ในกรณีที่มีตัวอย่างลบตัวแม้เพียงหนึ่งตัวที่สามารถแจงส่วนได้

จะพิจารณา γ_1, γ_2 ว่าอยู่ในรูปของ $\gamma_1 = \gamma_3\ \gamma_4$, $\gamma_2 = u\gamma_3w\gamma_4$ โดยที่ $\alpha_3, \alpha_4 \in (V \cup T)^*$ และ $v,x \in T^+$ หรือไม่ ซึ่งไม่อยู่ในรูปของ $\alpha_1 = \alpha_3\alpha_4$, $\alpha_2 = \alpha_3v\alpha_4x$ ดังนั้นจึงไม่เข้ากรณีแรก

ถ้าในกรณีนี้ กรณีแรกไม่สามารถสร้างกฎใหม่ได้

ดังนั้นจึงเข้าสู่กรณีที่สอง

พิจารณา γ_1, γ_2 อยู่ในรูปของ $\gamma_1 = \gamma_3$, $\gamma_2 = u\gamma_3$

โดยที่ $\gamma_3 \in (V \cup T)^*$ และ $u \in T^+$

จะได้ว่าเราจะสร้างกฎวนซ้ำตามแบบที่ 1 คือ

$S \longrightarrow ccAbb$

$A \longrightarrow Aaa \mid \lambda$

$A \longrightarrow aAa$

จากนั้นนำกฎวนซ้ำที่ได้ไปสร้างเป็นไวยากรณ์สมมติฐานแล้วนำไปทดสอบกับตัวอย่างลบทั้งหมด ถ้าไม่มีตัวอย่างลบตัวใดที่สามารถแจงส่วนได้จะทำการเปลี่ยนกฎเป็น

$\{S \longrightarrow ccAbb, A \longrightarrow Aaa, A \longrightarrow aAa, A \longrightarrow \lambda \} \cup TG - \{ S \longrightarrow ccabbb \}$

เมื่อทำการสร้างกฎใหม่ได้แล้วจึงจบการทำงานของขั้นตอนที่ 2

ในกรณีที่มีตัวอย่างลบตัวแม้เพียงหนึ่งตัวที่สามารถแจงส่วนได้ จะทำการสร้างกฎวนซ้ำตามแบบที่ 2 คือ

$S \longrightarrow ccAbb$

$A \longrightarrow aAa \mid \lambda$

จากนั้นนำกฎวนซ้ำที่ได้ไปสร้างเป็นไวยากรณ์สมมติฐานแล้วนำไปทดสอบกับตัวอย่างลบทั้งหมด ถ้าไม่มีตัวอย่างลบตัวใดที่สามารถแจงส่วนได้จะทำการเปลี่ยนกฎเป็น

$\{S \longrightarrow ccAbb, A \longrightarrow aAa, A \longrightarrow \lambda \} \cup TG - \{ S \longrightarrow ccabbb \}$

เมื่อทำการสร้างกฎใหม่ได้แล้วจึงจบการทำงานของขั้นตอนที่ 2

ในกรณีที่มีตัวอย่างลบตัวแม้เพียงหนึ่งตัวที่สามารถแจงส่วนได้ จะทำการสร้างกฎวนซ้ำตามแบบที่ 3 คือ

$S \longrightarrow ccAbb$

$A \longrightarrow Aaa \mid \lambda$

จากนั้นนำกฎวนซ้ำที่ได้ไปสร้างเป็นไวยากรณ์สมมติฐานแล้วนำไปทดสอบกับตัวอย่างลบทั้งหมด ถ้าไม่มีตัวอย่างลบตัวใดที่สามารถแจงส่วนได้จะทำการเปลี่ยนกฎเป็น

$\{ S \longrightarrow ccAbb, A \longrightarrow Aaa, A \longrightarrow \lambda \} \cup TG - \{ S \longrightarrow ccabbb \}$

เมื่อทำการสร้างกฎใหม่ได้แล้วจึงจบการทำงานของขั้นตอนที่ 2

ในกรณีที่มีตัวอย่างลบตัวแม้เพียงหนึ่งตัวที่สามารถแจงส่วนได้ จะไม่ทำการเปลี่ยนใดๆ

□

ขั้นตอนที่ 3 ถ้าไม่สามารถหากฎใหม่ได้ได้จากขั้นตอนที่ 1 และ ขั้นตอนที่ 2 จะแบ่งพิจารณาออกเป็น 2 กรณีคือ

กรณีที่ 1 ถ้า $\alpha_1 = \alpha_3 u$, $\alpha_2 = \alpha_3 v$ และ $\gamma_1 = w\gamma_3$, $\gamma_2 = x\gamma_3$

โดยที่ $\alpha_3, \gamma_3 \in (V \cup T)^*$ และ $u,w \in T^*$ $v,x \in T^+$

เราจะสร้างกฎวนซ้ำได้เป็น

$S \longrightarrow \alpha_3 A \gamma_3$

$A \longrightarrow vAx \mid uAw \mid \beta$

โดยที่ ตัวแปร A เป็นตัวแปรใหม่ที่ไม่ซ้ำกับตัวแปรที่มีอยู่ในไวยากรณ์

จากนั้นนำกฎวนซ้ำที่ได้ไปทดสอบการแจงส่วนตัวอย่างลบทุกตัวพร้อมกับกฎที่มีอยู่ก่อน ผลที่ได้จากการทดสอบการแจงส่วนกับตัวอย่างลบ จะบอกถึงกฎวนซ้ำที่เพิ่มเข้ามาใหม่ว่าสามารถเพิ่มเข้าไปในกฎที่มีอยู่ก่อนได้ ก็ต่อเมื่อไม่สามารถทำการแจงส่วนกับตัวอย่างลบทุกตัวได้ ดังนั้นเราจะทำการเปลี่ยนกฎเป็น

$$\{ S \rightarrow \alpha_3 A \gamma_3, A \rightarrow vAx, A \rightarrow \beta \} \cup TG - \{S \rightarrow S_{short}\}$$

กรณีที่ 2 ถ้าไม่เข้ากรณีแรกจะไม่ทำการเปลี่ยนแปลงใดๆ

ตัวอย่าง 3.5 *กำหนดให้* $S_{short} = abcd$, $S_{long} = aabbcd$

วิธีทำ

จากการเปรียบเทียบระหว่าง S_{short}, S_{long} จะได้ว่า

$\alpha_1 = \lambda$, $\alpha_2 = a$ และ $\gamma_1 = cd$, $\gamma_2 = bcd$, $\beta = ab$

ดังนั้นจะเราสร้างกฎวนซ้ำได้ดังนี้

$S \rightarrow Acd$

$A \rightarrow aAb \mid ab$

จากนั้นนำกฎวนซ้ำที่ได้ไปสร้างเป็นไวยากรณ์สมมติฐานแล้วนำไปทดสอบกับตัวอย่างลบทั้งหมด ถ้าไม่มีตัวอย่างลบตัวใดที่สามารถแจงส่วนได้จะทำการเปลี่ยนกฎเป็น

$$\{ S \rightarrow Acd, A \rightarrow aAb, A \rightarrow ab \} \cup TG - \{ S \rightarrow aabcd \}$$

เมื่อทำการสร้างกฎใหม่ได้แล้วจึงจบการทำงานของขั้นตอนที่ 3

ในกรณีที่มีตัวอย่างลบตัวแม้เพียงหนึ่งตัวที่สามารถแจงส่วนได้ จะไม่ทำการเปลี่ยนแปลง

□

ขั้นตอนที่ 4 ถ้าไม่สามารถสร้างกฎใหม่ได้จากทั้งสามขั้นตอนข้างต้นเราจะไปทำการพิจารณาคู่ของ S_{short}, S_{long} ถัดไป แต่ถ้าไม่มี S_{short}, S_{long} คู่ถัดไปจะไปสู่ขั้นตอนที่ 5

ขั้นตอนที่ 5 สุดท้ายถ้าไม่สามารถสร้างกฎใหม่ได้จากเปรียบเทียบทุกคู่ของ S_{short} และ S_{long} พิจารณาการแทนสายอักขระ S_{long} ด้วยกฎวนซ้ำที่สร้างขึ้นจากสายอักขระย่อยของ S_{long} ดังนี้

$$A_i \rightarrow A_i \alpha_i \mid \alpha_i$$

โดยที่ α_i เป็นสายอักขระย่อยทั้งหมดที่เป็นไปได้ของ S_{long} และ A_i เป็นตัวแปรใหม่ที่ไม่ซ้ำกับตัวแปรที่มีอยู่ในไวยากรณ์ และไม่มี $B \in V$ ใดๆที่ $B \rightarrow B\alpha_i$, $B \rightarrow \alpha_i \in P$

นำ $A_i \longrightarrow A_i\alpha_i \mid \alpha_i$ ที่สร้างขึ้นไปแทนในสายอักขระ S_{long} ด้วยอัลกอริทึมการแทนสายอักขระด้วยตัวแปร โดยทำเลือกทำการแทนตัวแปรด้วย A_i ที่สร้างขึ้นเท่านั้น ลำดับของการแทนจะเริ่มจาก $|\alpha_i|$ ที่น้อยไปหามาก ถ้า A_i ใดสามารถแทนในสายอักขระ S_{long} ได้จะทำการเพิ่มกฎ $\{A_i \longrightarrow A_i\alpha_i , A_i \longrightarrow \alpha_i\} \cup TG$ หลังจากการแทน S_{long} ด้วยกฎวนซ้ำที่สร้างขึ้นจากสายอักขระย่อยของ S_{long} ให้ S_{long} ใหม่ที่ได้ แทนด้วย S'_{long} สุดท้ายทำการเพิ่มกฎ

$$\{S \longrightarrow S'_{long}\} \cup TG$$

จากขั้นตอนทั้งหมดเราจะได้ไวยากรณ์ที่สามารถสร้างตัวอย่างบวกที่กำลังพิจารณาและตัวอย่างอย่างบวกก่อนหน้าได้ทั้งหมดซึ่งขั้นตอนที่ 1-3 สามารถสลับขั้นตอนการทำงานกันได้ เนื่องจากเป็นขั้นตอนการหาไวยากรณ์ทุกแบบที่สามารถเกิดขึ้นได้จากการเปรียบเทียบสายอักขระสองสาย แต่การสลับขั้นตอนอาจจะทำให้ได้ไวยากรณ์สุดท้ายที่แตกต่างกัน เนื่องจากไวยากรณ์สามารถเขียนบรรยายได้หลายรูปแบบ

3.6 อัลกอริทึมการผสานตัวแปรใหม่

อัลกอริทึมการผสานตัวแปรใหม่ คือ อัลกอริทึมที่ทำหน้าที่ทดสอบการผสานตัวแปรที่สร้างขึ้นมาใหม่ว่าสามารถรวมกับกฎเก่า หรือลด λ ออกจากตัวแปรใหม่ได้หรือไม่ ซึ่งประกอบด้วยขั้นตอนการทำงานดังนี้

ขั้นตอนที่ 1 ทำการพิจารณากฎ $S \longrightarrow \beta$ ที่สร้างขึ้นจากใหม่ทุกกฎ โดยที่ $\beta \in (V \cup T)^*$

แบบที่ 1 ถ้า β มีสายอักขระย่อยที่ปรากฏในรูป αA หรือ $A\alpha$ โดยที่ $A \in V$ และ $\alpha \in (V \cup T)^*$ เป็นตัวแปรที่สร้างใหม่และมีกฎในลักษณะ $A \longrightarrow A\alpha$, $A \longrightarrow \lambda$ แล้วเราจะพิจารณาเปลี่ยนกฎวนซ้ำจาก

$$S \longrightarrow \gamma_1\alpha A\gamma_2$$

หรือ

$$S \longrightarrow \gamma_1 A\alpha\gamma_2$$

เป็น

$$S \longrightarrow \gamma_1 A\gamma_2$$

โดยที่ $\gamma_1, \gamma_2 \in (V \cup T)^*$, $\beta = \gamma_1\alpha A\gamma_2$ หรือ $\beta = \gamma_1 A\alpha\gamma_2$

ทำการเปลี่ยนกฎ $A \longrightarrow \lambda$ เป็น $A \longrightarrow \alpha$ ทำเช่นนี้กับทุกกฎใหม่ที่ทางด้านซ้ายมือเท่ากับ S

ถ้าไม่มีสายอักขระย่อยปรากฏในรูป αA หรือ $A\alpha$ โดยที่มีแค่ A เป็นสายอักขระย่อย $S \longrightarrow \gamma_1 A\gamma_2$ และมีกฎในลักษณะ $A \longrightarrow A\alpha$, $A \longrightarrow \lambda$ แล้วเราจะทำการเพิ่มกฎ

$$\{S \longrightarrow \gamma_1\gamma_2\} \cup TG$$

แบบที่ 2 ถ้า β มีสายอักขระย่อยที่ปรากฏในรูป AB โดยที่ A, B $\in$ V จะได้ว่าเราจะพิจารณาเปลี่ยนกฎวนซ้ำจาก

$$S \rightarrow \alpha AB\gamma$$

เปลี่ยนเป็น

$$S \rightarrow \alpha E\gamma$$

$$E \rightarrow EA \mid EB \mid A \mid B$$

โดยที่ $\alpha, \gamma \in (V \cup T)^*$, $\beta = \alpha AB\gamma$ ตัวแปร E เป็นตัวแปรใหม่ที่ไม่ซ้ำกับตัวแปรที่มีอยู่ในไวยากรณ์

จากนั้นนำกฎวนซ้ำที่ได้ไปทดสอบการแจงส่วนตัวอย่างลบทุกตัวพร้อมกับกฎที่มีอยู่ก่อน ผลที่ได้จากการทดสอบการแจงส่วน จะบอกถึงกฎการวนซ้ำที่เปลี่ยนไปว่าสามารถเปลี่ยนให้เข้ากับกฎที่มีอยู่ก่อนได้ ก็ต่อเมื่อไม่สามารถทำการแจงส่วนกับตัวอย่างลบทุกตัวได้ ดังนั้นเราจะทำการเปลี่ยนกฎเป็น

$$\{ S \rightarrow \alpha E\gamma, E \rightarrow EA, E \rightarrow EB, E \rightarrow A, E \rightarrow B \} \cup TG - \{ S \rightarrow \alpha AB\gamma \}$$

ตัวอย่าง 3.6 *กำหนดให้กฎที่สร้างใหม่ คือ* $P' = \{S \rightarrow aBcA, A \rightarrow Ac, A \rightarrow \lambda, B \rightarrow Ba, B \rightarrow a\}$

วิธีทำ

พิจารณากฎใหม่ที่ทางด้านซ้ายมีคือตัวแปรเริ่มต้น S $\rightarrow$ aBcA ดังนั้นเราจะสามารถลดกฎ A $\rightarrow \lambda$ ได้ตามแบบที่ 1 โดยเปลี่ยนกฎ S $\rightarrow$ aBcA ได้ดังนี้

$$\{S \rightarrow aBA, A \rightarrow Ac, A \rightarrow c\}$$

พิจารณากฎที่สร้างใหม่อีกครั้งจะได้ว่า S $\rightarrow$ aBA สามารถเปลี่ยนกฎวนซ้ำได้ตามแบบ 2 โดยเปลี่ยนกฎ S $\rightarrow$ aBA เป็น

$$S \rightarrow aE$$

$$E \rightarrow EA \mid EB \mid A \mid B$$

จากนั้นนำไปทดสอบการแจงส่วนกับตัวอย่างลบทุกตัว ถ้าไม่สามารถแจงส่วนได้ตัวอย่างลบทุกตัวเราจะเปลี่ยนกฎเป็น

$$\{S \rightarrow aE, E \rightarrow EA, E \rightarrow EB, E \rightarrow A, E \rightarrow B, A \rightarrow Ac, A \rightarrow c\}$$

แต่ถ้าสามารถแจงส่วนได้ตัวอย่างลบแม้เพียงตัวเดียวจะไม่ทำการเปลี่ยนแปลง

□

ขั้นตอนที่ 2 พิจารณาแต่ละกฎใหม่ นำไปเปรียบเทียบกับแต่ละกฎที่มีอยู่ก่อนหน้า ถ้าการเปรียบเทียบกฎอยู่ในรูปแบบดังนี้ $S \rightarrow \alpha B\gamma$ เป็นกฎที่อยู่ในกฎเก่าและ $S \rightarrow \alpha C\gamma$ เป็นกฎที่อยู่ในกฎใหม่ โดยที่ $\alpha, \gamma \in (V \cup T)^*$ และ $B,C \in V$ และมีกฎ $B \rightarrow uBv$, $B \rightarrow \beta_1$ และ $C \rightarrow xCy$, $C \rightarrow \beta_2 \in P$ โดยที่ $u,v,x,y \in T^+$, $\beta \in (V \cup T)^*$ จะทำการสร้างกฎวนซ้ำได้เป็น

$$S \rightarrow \alpha E\gamma$$

$$E \rightarrow uEv \mid xEy \mid EB \mid EC \mid B \mid C$$

โดยที่ ตัวแปร E เป็นตัวแปรใหม่ที่ไม่ซ้ำกับตัวแปรที่มีอยู่ในไวยากรณ์

จากนั้นนำกฎวนซ้ำที่ได้ไปทดสอบการแจงส่วนตัวอย่างลบทุกตัวพร้อมกับกฎที่มีอยู่ก่อน ผลที่ได้จากการทดสอบการแจงส่วน จะบอกถึงกฎการวนซ้ำที่เพิ่มเข้ามาใหม่ว่าสามารถเพิ่มเข้าไปในกฎที่มีอยู่ก่อนได้ ก็ต่อเมื่อไม่สามารถทำการแจงส่วนกับตัวอย่างลบทุกตัวได้ ดังนั้นเราจะทำการเพิ่มกฎเป็น

$$\{ S \rightarrow \alpha E\gamma, E \rightarrow uEv, E \rightarrow xEy, E \rightarrow EB, E \rightarrow EC, E \rightarrow B, E \rightarrow C \} \cup TG - \{ S \rightarrow \alpha B\gamma, S \rightarrow \alpha C\gamma \}$$

ถ้าสามารถแจงส่วนได้เป็นตัวอย่างลบแม้เพียงตัวเดียวจะพิจารณาสร้างกฎวนซ้ำเป็น

$$S \rightarrow \alpha E\gamma$$

$$E \rightarrow uEv \mid xEy \mid B \mid C$$

จากนั้นนำกฎวนซ้ำที่ได้ไปทดสอบการแจงส่วนตัวอย่างลบทุกตัวพร้อมกับกฎที่มีอยู่ก่อน ผลที่ได้จากการทดสอบการแจงส่วน จะบอกถึงกฎการวนซ้ำที่เพิ่มเข้ามาใหม่ว่าสามารถเพิ่มเข้าไปในกฎที่มีอยู่ก่อนได้ ก็ต่อเมื่อไม่สามารถทำการแจงส่วนกับตัวอย่างลบทุกตัวได้ ดังนั้นเราจะทำการเพิ่มกฎเป็น

$$\{ S \rightarrow \alpha E\gamma, E \rightarrow uEv, E \rightarrow xEy, E \rightarrow B, E \rightarrow C \} \cup TG - \{ S \rightarrow \alpha B\gamma, S \rightarrow \alpha C\gamma \}$$

ถ้าสามารถแจงส่วนได้เป็นตัวอย่างลบแม้เพียงตัวเดียวจะพิจารณาสร้างกฎวนซ้ำเป็น

$$S \rightarrow \alpha E\gamma$$

$$E \rightarrow EB \mid EC \mid B \mid C$$

จากนั้นนำกฎวนซ้ำที่ได้ไปทดสอบการแจงส่วนตัวอย่างลบทุกตัวพร้อมกับกฎที่มีอยู่ก่อน ผลที่ได้จากการทดสอบการแจงส่วน จะบอกถึงกฎการวนซ้ำที่เพิ่มเข้ามาใหม่ว่าสามารถเพิ่มเข้าไปในกฎที่มีอยู่ก่อนได้ ก็ต่อเมื่อไม่สามารถทำการแจงส่วนกับตัวอย่างลบทุกตัวได้ ดังนั้นเราจะทำการเพิ่มกฎเป็น

$$\{ S \rightarrow \alpha E\gamma,\ E \rightarrow EB,\ E \rightarrow EC,\ E \rightarrow B,\ E \rightarrow C \} \cup TG - \{ S \rightarrow \alpha B\gamma,\ S \rightarrow \alpha C\gamma \}$$

ตัวอย่างที่ 3.7 *ให้ที่กฎสร้างใหม่ P' = {S $\rightarrow$ B, B $\rightarrow$ aBb, B $\rightarrow$ ab} ได้จากอัลกอริทึมการสร้างตัวแปรวนซ้ำโดยวิธีหาสายอักขระย่อยที่มีความยาวมากสุด มีกฎก่อนหน้าคือ P = { S $\rightarrow$ A, A $\rightarrow$ bAa, B $\rightarrow$ ab }*

วิธีทำ

พิจารณากฎใหม่ที่ทางด้านซ้ายมีคือตัวแปรเริ่มต้น S $\rightarrow$ B เทียบกับกฎเก่าที่ S $\rightarrow$ A และมี B $\rightarrow$ aBb, B $\rightarrow$ ab และ A $\rightarrow$ bAa, B $\rightarrow$ ab ดังนั้นเราจะสร้างกฎวนซ้ำได้เป็น

$$S \rightarrow C$$
$$C \rightarrow aCb \mid bCa \mid CA \mid CB \mid A \mid B$$

จากนั้นนำกฎวนซ้ำที่สร้างใหม่ไปทดสอบการแจงส่วนกับตัวอย่างลบทุกตัว ถ้าไม่สามารถแจงส่วนได้ตัวอย่างลบทุกตัวเราจะเพิ่มกฎเป็น

$$\{S \rightarrow C,\ C \rightarrow aAb,\ C \rightarrow bAa,\ C \rightarrow CA,\ C \rightarrow CB,\ C \rightarrow A,\ C \rightarrow B\} \cup TG - \{ S \rightarrow A,\ S \rightarrow B \}$$

ในกรณีที่สามารถแจงส่วนได้ตัวอย่างลบแม้เพียงตัวเดียวจะทำการสร้างกฎวนซ้ำได้เป็น

$$S \rightarrow C$$
$$C \rightarrow aCb \mid bCa \mid A \mid B$$

จากนั้นนำกฎวนซ้ำที่สร้างใหม่ไปทดสอบการแจงส่วนกับตัวอย่างลบทุกตัว ถ้าไม่สามารถแจงส่วนได้ตัวอย่างลบทุกตัวเราจะเพิ่มกฎเป็น

$$\{S \rightarrow C,\ C \rightarrow aAb,\ C \rightarrow bAa,\ C \rightarrow A,\ C \rightarrow B\} \cup TG - \{ S \rightarrow A,\ S \rightarrow B \}$$

ในกรณีที่สามารถแจงส่วนได้ตัวอย่างลบแม้เพียงตัวเดียวจะทำการสร้างกฎวนซ้ำได้เป็น

$$S \rightarrow C$$
$$C \rightarrow CA \mid CB \mid A \mid B$$

จากนั้นนำกฎวนซ้ำที่สร้างใหม่ไปทดสอบการแจงส่วนกับตัวอย่างลบทุกตัว ถ้าไม่สามารถแจงส่วนได้ตัวอย่างลบทุกตัวเราจะเพิ่มกฎเป็น

$\{S \rightarrow C, C \rightarrow CA, C \rightarrow CB, C \rightarrow A, C \rightarrow B\} \cup TG - \{S \rightarrow A, S \rightarrow B\}$

ในกรณีที่สามารถแจงส่วนได้ตัวอย่างลบแม้เพียงตัวเดียวจะไม่ทำการเปลี่ยนแปลง □

ตัวอย่างที่ 3.8 *กำหนดให้ภาษา* $L = \{ a^i b^j c^j \mid a,b,c \in \Sigma$ *และ* $i, j > 0\}$

S_P *เป็นเซตของสายอักขระ {abc, abcc, aabbc, abccc, aabbcc, aabbccc , aaabbbc }*

S_N *เป็นเซตของสายอักขระ {a, b, c, ab, ac, aab, abb, abbc, aabc, abcbc,* aabcc, *aaabcc, aaabbcccc}*

วิธีทำ

เริ่มต้น

รอบที่ 0 ทำการอ่าน abc จาก S_p ได้ $S_p[0]$ = “abc“ ซึ่งไม่สามารถทำการแจงส่วนได้ด้วย TG ปัจจุบัน

ทำการสร้างตัวแปรวนซ้ำจากสายอักขระย่อยของ abc จะได้

$TG = \{ S \rightarrow abA, A \rightarrow Ac, A \rightarrow c \}$

รอบที่ 1 ทำการอ่าน abcc จาก S_p ได้ $S_p[1]$ = “abcc“ ซึ่งสามารถทำการแจงส่วนได้ด้วย TG ปัจจุบันดังนั้นจึงไม่ทำการเปลี่ยนแปลงใดๆ

รอบที่ 2 ทำการอ่าน aabbc จาก S_p ได้ $S_p[2]$ = “aabbc“ ซึ่งไม่สามารถทำการแจงส่วนได้ด้วย TG ปัจจุบัน

ดังนั้นจึงไปสู่ขั้นตอนของการแทนสายอักขระด้วยตัวแปรซึ่งสามารถแปลง c เป็นA ได้ เมื่อทดสอบกับ SN แล้วพบว่าไม่สามารถแจกแจงตัวใดใน SN ได้ ดังนั้นจึงแทน c ด้วย A Sp[2] = “aabbA”

จากนั้นเข้าสู่ขั้นตอนของการหาสายอักขระย่อยที่ยาวที่สุด เราสามารถหาสายอักขระย่อยที่ยาวที่สุดที่มีในกฎทางขวาของ S ใน TG ได้คือ “ab”

เราจะได้ $\alpha_1 = \lambda$, α_2 = a, γ_1 = A, γ_2 = bA

พิจารณา $\alpha_1 = \lambda$, α_2 = a

ดังนั้นเราจะทำการสร้างกฎวนซ้ำจาก α_1, α_2 ได้เป็น

$S \rightarrow BabA$

$S \rightarrow BabbA$

$B \rightarrow Ba \mid a \mid \lambda$

นำกฎวนซ้ำที่สร้างขึ้นไปสร้างเป็นไวยากรณ์สมมติฐานเพื่อทดสอบกับ S_N จะได้ว่า สามารถทำการแจงส่วนไปยัง aaabcc ซึ่ง aaabcc เป็นสมาชิกของ S_N เพราะฉะนั้นจึงไม่สามารถเพิ่มกฎวนซ้ำไปใน TG ได้

พิจารณา γ_1 = A, γ_2 = bA

ดังนั้นเราจะทำการสร้างกฎวนซ้ำจาก γ_1, γ_2 ได้เป็น

$S \longrightarrow abB$

$S \longrightarrow aabB$

$B \longrightarrow BbA \mid BA \mid A \mid bA$

นำกฎวนซ้ำที่สร้างขึ้นไปสร้างเป็นไวยากรณ์สมมติฐานเพื่อทดสอบกับ S_N จะได้ว่า สามารถทำการแจงส่วนไปยัง abcbc ซึ่ง abcbc เป็นสมาชิกของ S_N เพราะฉะนั้นจึงไม่สามารถเพิ่มกฎวนซ้ำไปใน TG ได้

พิจารณา γ_1 = A, γ_2 = bA

ดังนั้นเราจะทำการสร้างกฎวนซ้ำจากกรณีที่ 2 ของ γ_1, γ_2 ได้เป็น

$S \longrightarrow abBA$

$S \longrightarrow aabBA$

$B \longrightarrow Bb \mid b \mid \lambda$

นำกฎวนซ้ำที่สร้างขึ้นไปสร้างเป็นไวยากรณ์สมมติฐานเพื่อทดสอบกับ S_N จะได้ว่า สามารถทำการแจงส่วนไปยัง aabcc ได้ ซึ่ง aabcc เป็นสมาชิกของ S_N เพราะฉะนั้นจึงไม่สามารถเพิ่มกฎวนซ้ำไปใน TG ได้

พิจารณา α_1 = λ, α_2 = a และ γ_1 = A, γ_2 = bAไปพร้อมกันตามขั้นตอนที่ 3

ดังนั้นเราจะทำการสร้างกฎ

$S \longrightarrow BA$

$B \longrightarrow aBb \mid ab$

นำไปสร้างเป็นไวยากรณ์สมมติฐานเพื่อทดสอบกับ S_N จะได้ว่า ไม่สามารถแจงส่วนไปสมาชิกทั้งหมดของ S_N เพราะฉะนั้นจึงสามารถเพิ่มกฎวนซ้ำไปใน TG ได้

เปลี่ยน TG = { $S \longrightarrow BA$, $A \longrightarrow Ac$, $A \longrightarrow c$, $B \longrightarrow aBb$, $B \longrightarrow ab$ }

จากนั้นทำเข้าสู่ขั้นตอนการลดตัวแปรของกฎที่สร้างขึ้นจะได้ว่า

TG = { $S \longrightarrow BA$, $A \longrightarrow Ac$, $A \longrightarrow c$, $B \longrightarrow aBb$, $B \longrightarrow ab$ }

เมื่อ TG = { S $\longrightarrow$ BA , A $\longrightarrow$ Ac, A $\longrightarrow$ c, B $\longrightarrow$ aBb, B $\longrightarrow$ ab } แล้ว จะได้ว่า TG สามารถแจกแจงตัวอย่างบวกที่เหลือได้ แต่ไม่สามารถแจกแจงตัวอย่างลบได้ ดังนั้น กฎของไวยากรณ์ไม่พึ่งบริบทที่ได้จากการอัลกอริทึมคือ

S $\longrightarrow$ BA

A $\longrightarrow$ Ac | c

B $\longrightarrow$ aBb | ab

□

ตัวอย่างที่ 3.9 *กำหนดให้ภาษา L = { w | w* $\in \Sigma$ *,* Σ *= {a, b}, จำนวนของ a = จำนวนของ b}*

S_P *เป็นเซตของสายอักขระ {ab, ba, aabb, bbaa, aaabbb, aababb , baabba}*

S_N *เป็นเซตของสายอักขระ {a, b, aab, abb, aba, baa, bba, aaab, abbba }*

วิธีทำ

เริ่มต้น

รอบที่ 0 ทำการอ่าน ab จาก S_p ได้ $S_p[0]$ = “ab“ ซึ่งไม่สามารถทำการแจงส่วนได้ด้วย TG ปัจจุบัน

ทำการสร้างตัวแปรวนซ้ำจากสายอักขระย่อยของ ab จะได้

TG = { S $\longrightarrow$ A, A $\longrightarrow$ Aab, A $\longrightarrow$ ab }

รอบที่ 1 ทำการอ่าน ba จาก S_p ได้ $S_p[1]$ = “ba“ ซึ่งไม่สามารถทำการแจงส่วนได้ด้วย TG ปัจจุบัน

ดังนั้นจึงไปสู่ขั้นตอนของการแทนสายอักขระด้วยตัวแปรซึ่งไม่มีสายอักขระใดที่สามารถแปลงเป็นตัวแปรได้

จากนั้นเข้าสู่ขั้นตอนของการหาสายอักขระย่อยที่ยาวที่สุด ซึ่งไม่สามารถหาสายอักขระย่อยที่ยาวที่สุดที่มีในกฎทางขวาของ S ใน TG ได้ ดังนั้นจะทำการสร้างตัวแปรวนซ้ำจากสายอักขระย่อยของ ba จะได้ว่า

{S $\longrightarrow$ B, B $\longrightarrow$ Bba, B $\longrightarrow$ ba } $\cup$ TG

จากนั้นทำเข้าสู่ขั้นตอนการลดตัวแปรของกฎที่สร้างขึ้นจะได้ว่า

TG = { S $\longrightarrow$ B, B $\longrightarrow$ Bba, B $\longrightarrow$ ba, S $\longrightarrow$ A, A $\longrightarrow$ Aab, A $\longrightarrow$ ab }

รอบที่ 2 ทำการอ่าน aabb จาก S_p ได้ $S_p[2]$ = “aabb“ ซึ่งไม่สามารถทำการแจงส่วนได้ด้วย TG ปัจจุบัน

ดังนั้นจึงไปสู่ขั้นตอนของการแทนสายอักขระด้วยตัวแปรซึ่งเราสามารถแทนสายอักขระได้ด้วย A ดังนั้นเราจะแทน aabb ด้วย aAb

จากนั้นเข้าสู่ขั้นตอนของการหาสายอักขระย่อยที่ยาวที่สุด เราสามารถหาสายอักขระย่อยที่ยาวที่สุดที่มีในกฎทางขวาของ S ใน TG ได้คือ “A”
เราจะได้ $\alpha_1 = \lambda$, $\alpha_2 = a$, $\gamma_1 = \lambda$, $\gamma_2 = b$
พิจารณา $\alpha_1 = \lambda$, $\alpha_2 = a$
ดังนั้นเราจะทำการสร้างกฎวนซ้ำจาก α_1, α_2 ได้เป็น

$S \longrightarrow CA$

$S \longrightarrow CAb$

$C \longrightarrow Ca \mid a \mid \lambda$

นำกฎวนซ้ำที่สร้างขึ้นไปสร้างเป็นไวยากรณ์สมมติฐานเพื่อทดสอบกับ S_N จะได้ว่า สามารถทำการแจงส่วนไปยัง aaab ซึ่ง aaab เป็นสมาชิกของ S_N เพราะฉะนั้นจึงไม่สามารถเพิ่มกฎวนซ้ำไปใน TG ได้
พิจารณา $\gamma_1 = \lambda$, $\gamma_2 = b$
ดังนั้นเราจะทำการสร้างกฎวนซ้ำจาก γ_1, γ_2 ได้เป็น

$S \longrightarrow AC$

$S \longrightarrow aAC$

$C \longrightarrow Cb \mid \lambda \mid b$

นำกฎวนซ้ำที่สร้างขึ้นไปสร้างเป็นไวยากรณ์สมมติฐานเพื่อทดสอบกับ S_N จะได้ว่า จะได้ว่า สามารถทำการแจงส่วนไปยัง aab ซึ่ง aab เป็นสมาชิกของ S_N เพราะฉะนั้นจึงไม่สามารถเพิ่มกฎวนซ้ำไปใน TG ได้

พิจารณา $\alpha_1 = \lambda$, $\alpha_2 = a$ และ $\gamma_1 = \lambda$, $\gamma_2 = b$ไปพร้อมกันตามขั้นตอนที่ 3
ดังนั้นเราจะทำการสร้างกฎ

$S \longrightarrow C$

$C \longrightarrow aCb \mid A$

นำกฎวนซ้ำที่สร้างขึ้นไปสร้างเป็นไวยากรณ์สมมติฐานเพื่อทดสอบกับ S_N จะได้ว่า ไม่สามารถแจงส่วนไปสมาชิกทั้งหมดของ S_N เพราะฉะนั้นจึงสามารถเพิ่มกฎวนซ้ำไปใน TG ได้
ดังนั้นเราจะทำการเพิ่มกฎ $\{S \longrightarrow C, C \longrightarrow aCb, C \longrightarrow A\} \cup TG - \{S \longrightarrow A\}$

จากนั้นทำเข้าสู่ขั้นตอนการลดตัวแปรของกฎที่สร้างขึ้นจะได้ว่า

$TG = \{ S \rightarrow C, C \rightarrow aCb, C \rightarrow A, S \rightarrow B, B \rightarrow Bba, B \rightarrow ba, A \rightarrow Aab, A \rightarrow ab \}$

รอบที่ 3 ทำการอ่าน bbaa จาก S_p ได้ $S_p[3]$ = “bbaa“ ซึ่งไม่สามารถทำการแจงส่วนได้ด้วย TG ปัจจุบัน

ดังนั้นจึงไปสู่ขั้นตอนของการแทนสายอักขระด้วยตัวแปรซึ่งเราสามารถแทนสายอักขระได้ด้วย B ดังนั้นเราจะแทน bbaa ด้วย bBa

จากนั้นเข้าสู่ขั้นตอนของการหาสายอักขระย่อยที่ยาวที่สุด เราสามารถหาสายอักขระย่อยที่ยาวที่สุดที่มีในกฎทางขวาของ S ใน TG ได้คือ “B”

เราจะได้ $\alpha_1 = \lambda$, $\alpha_2 = b$, $\gamma_1 = \lambda$, $\gamma_2 = a$

พิจารณา $\alpha_1 = \lambda$, $\alpha_2 = a$

ดังนั้นเราจะทำการสร้างกฎวนซ้ำจาก α_1, α_2 ได้เป็น

$S \rightarrow DB$

$S \rightarrow DBa$

$D \rightarrow Db \mid b \mid \lambda$

นำกฎวนซ้ำที่สร้างขึ้นไปสร้างเป็นไวยากรณ์สมมติฐานเพื่อทดสอบกับ S_N จะได้ว่า สามารถทำการแจงส่วนไปยัง bba ซึ่ง bba เป็นสมาชิกของ S_N เพราะฉะนั้นจึงไม่สามารถเพิ่มกฎวนซ้ำไปใน TG ได้

พิจารณา $\gamma_1 = \lambda$, $\gamma_2 = a$

ดังนั้นเราจะทำการสร้างกฎวนซ้ำจาก γ_1, γ_2 ได้เป็น

$S \rightarrow BD$

$S \rightarrow bBD$

$D \rightarrow Da \mid \lambda \mid a$

นำกฎวนซ้ำที่สร้างขึ้นไปสร้างเป็นไวยากรณ์สมมติฐานเพื่อทดสอบกับ S_N จะได้ว่า จะได้ว่า สามารถทำการแจงส่วนไปยัง bba ซึ่ง bba เป็นสมาชิกของ S_N เพราะฉะนั้นจึงไม่สามารถเพิ่มกฎวนซ้ำไปใน TG ได้

พิจารณา $\alpha_1 = \lambda$, $\alpha_2 = a$ และ $\gamma_1 = \lambda$, $\gamma_2 = b$ไปพร้อมกันตามขั้นตอนที่ 3

ดังนั้นเราจะทำการสร้างกฎ

$S \rightarrow D$

$D \rightarrow bDa \mid B$

นำกฎวนซ้ำที่สร้างขึ้นไปสร้างเป็นไวยากรณ์สมมติฐานเพื่อทดสอบกับ S_N จะได้ว่า ไม่สามารถแจงส่วนไปสมาชิกทั้งหมดของ S_N เพราะฉะนั้นจึงสามารถเพิ่มกฎวนซ้ำไปใน TG ได้

จะได้ว่าเราจะทำการเพิ่มกฎ $\{S \rightarrow D, D \rightarrow bDa, D \rightarrow B\} \cup TG - \{S \rightarrow B\}$

$TG = \{ S \rightarrow D, D \rightarrow bDa, D \rightarrow B, S \rightarrow C, C \rightarrow aCb,$

$C \rightarrow A, B \rightarrow Bba, B \rightarrow ba, A \rightarrow Aab, A \rightarrow ab \}$

จากนั้นทำเข้าสู่ขั้นตอนการลดตัวแปรของกฎที่สร้างขึ้นจะได้ว่า เราจะทำการสร้างกฎ

$S \rightarrow E$

$E \rightarrow aEb \mid bEa \mid EC \mid ED \mid C \mid D$

นำไปสร้างเป็นไวยากรณ์สมมติฐานเพื่อทดสอบกับ S_N จะได้ว่า ไม่สามารถแจงส่วนไปสมาชิกทั้งหมดของ S_N เพราะฉะนั้นจึงสามารถเพิ่มกฎวนซ้ำไปใน TG ได้

จะได้ว่าเราจะทำการเพิ่มกฎ $\{S \rightarrow E, E \rightarrow aEb, E \rightarrow bEa,$

$E \rightarrow EC, E \rightarrow ED, E \rightarrow C, E \rightarrow D\} \cup TG - \{S \rightarrow C,$

$S \rightarrow D\}$

ดังนั้น $TG = \{ S \rightarrow E, E \rightarrow aEb, E \rightarrow bEa, E \rightarrow EC,$

$E \rightarrow ED, E \rightarrow C, E \rightarrow D, C \rightarrow aCb, C \rightarrow A, D \rightarrow bDa,$

$D \rightarrow B, B \rightarrow Bba, B \rightarrow ba, A \rightarrow Aab, A \rightarrow ab \}$

เมื่อ $TG = \{ S \rightarrow E, E \rightarrow aEb, E \rightarrow bEa, E \rightarrow EC, E \rightarrow ED,$

$E \rightarrow C, E \rightarrow D, C \rightarrow aCb, C \rightarrow A, D \rightarrow bDa, D \rightarrow B,$

$B \rightarrow Bba, B \rightarrow ba, A \rightarrow Aab, A \rightarrow ab \}$

จะได้ว่า TG สามารถแจกแจงตัวอย่างบวกที่เหลือได้ แต่ไม่สามารถแจกแจงตัวอย่างลบได้ ดังนั้น กฎของไวยากรณ์ไม่พึ่งบริบทที่ได้จากการอัลกอริทึมคือ

$S \rightarrow E$

$E \rightarrow aEb \mid bEa \mid EC \mid ED \mid C \mid D$

$C \rightarrow aCb \mid A$

$D \rightarrow bDa \mid B$

$A \rightarrow Aab \mid ab$

$B \rightarrow Bba \mid ba$ □

3.7 ผลการวิเคราะห์

เนื่องจากมีทฤษฎีเกี่ยวกับภาษาสม่ำเสมอซึ่งที่มีการพิสูจน์ไว้แล้วว่า ภาษาสม่ำเสมอจะมีคุณสมบัติดังทฤษฎีบทที่ 2.1 และจากการวิเคราะห์อัลกอริทึมและคุณสมบัติของภาษาสม่ำเสมอเราได้เสนอนิยามและผลวิเคราะห์สรุปเป็นทฤษฎีดังนี้

นิยามที่ 3.1 *S_P เป็นเซตของตัวอย่างบวกที่มีโครงสร้างสมบูรณ์ของภาษาสม่ำเสมอ L โดยที่ L ยอมรับได้โดยออโตมาตา ที่มีจำนวนสถานะเท่ากับ n ก็ต่อเมื่อ ทุก x ที่เป็นสมาชิกของ L ถ้า $|x| \leq 2n$ แล้ว x ต้องเป็นสมาชิกของ S_Pด้วย*

นิยามที่ 3.2 *S_N เป็นเซตของตัวอย่างลบที่มีโครงสร้างสมบูรณ์ของภาษาสม่ำเสมอ L โดยที่ L ยอมรับได้โดยออโตมาตา ที่มีจำนวนสถานะเท่ากับ n ก็ต่อเมื่อ ทุก x ที่ไม่ได้เป็นสมาชิกของ L ถ้า $|x| \leq 2n$ แล้ว x ต้องเป็นสมาชิกของ S_N ด้วย*

ทฤษฎีบทที่ 3.1 *ถ้าภาษา L เป็นภาษาสม่ำเสมอ และ S_P เป็นตัวอย่างบวกที่มีโครงสร้างสมบูรณ์ และ S_N เป็นตัวอย่างลบที่มีโครงสร้างสมบูรณ์แล้ว อัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบทสามารถจำแนกโดยจำกัดได้ทุก L*

พิสูจน์

สมมติให้ L เป็นภาษาสม่ำเสมอ จะต้องมีออโตมาตาที่มีจำนวนสถานะ n ดั้งนั้นย่อมมีสายอักขระที่ยอมรับโดยออโตมาตาที่มีความยาวไม่เกิน n โดยที่ไม่เกิดการวนซ้ำขึ้น ดังนั้นสายอักขระที่ยาวกว่า n ต้องเกิดการวนซ้ำขึ้นและการวนซ้ำหนึ่งครั้งจะมีความยาวไม่เกิน n ทำให้เกิดสายอักขระที่มีความยาวไม่เกิน 2n

เนื่องจากหลักการทำงานของอัลกอริทึมการสร้างตัวแปรวนซ้ำขั้นตอนที่ 5 จะทำการพิจารณาตัวอย่างบวกที่เข้ามา แล้วสร้างตัวแปรวนซ้ำทุกตัวที่เกิดขึ้นได้ในสายอักขระที่เป็นสายอักขระจากตัวอย่างบวกที่มีโครงสร้างสมบูรณ์ และตัวแปรวนซ้ำที่สร้างขึ้นจะถูกจำกัดโดยตัวอย่างลบที่มีโครงสร้างสมบูรณ์ ดังนั้นจากทฤษฎีบทที่ 2.1 ถ้าเราทดสอบการสร้างตัวแปรวนซ้ำทุกแบบที่เกิดขึ้นได้แล้วจะได้ว่าเราจะสามารถค้นพบรูปแบบวนซ้ำของไวยากรณ์สม่ำเสมอที่สามารถสร้างสายอักขระใด ๆที่มีความยาวมากกว่า 2n และเป็นสมาชิกของ L ได้เสมอ

■

ทฤษฎีบทที่ 3.2 *อัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบทสามารถอนุมานไวยากรณ์ไม่พึ่งบริบท โดยใช้ความซับซ้อนเชิงเวลาที่ไม่เกินฟังก์ชันพหุนามตามผลบวกรวมของความยาวแต่ละตัวอย่างบวก($||S_P||$) และผลบวกของความยาวรวมแต่ละตัวอย่างลบ ($||S_N||$)*

พิสูจน์

เนื่องจากอัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบทเริ่มการทำงานจากตัวอย่างบวกทีละตัว และทำการพิจารณาเป็นแต่ละอัลกอริทึมย่อยเป็นขั้นตอนดังนั้นเราต้องแยกการพิจารณาออกเป็นแต่ละอัลกอริทึมแล้วจึงนำมาหาความซับซ้อนรวมอีกครั้ง

การพิจารณาอัลกอริทึมการแทนสายอักขระด้วยตัวแปร เนื่องจากอัลกอริทึมทำงานด้วยการแทนสายอักขระสมาชิกใน S_P ด้วยตัวแปรที่มีอยู่ก่อนหน้าดังนั้นจำนวนรอบของการแทนสายอักขระทั้งหมดที่เป็นไปจะขึ้นอยู่กับจำนวนชั้นของกฎวนซ้ำที่มีอยู่ ถ้าให้จำนวนชั้นของกฎวนซ้ำที่มีอยู่เป็น k ความยาวของสายอักขระที่กำลังพิจารณาเป็น n และกรณีเลวร้ายที่สุดคือในแต่ละรอบไม่สามารถแทนทุกสายอักขระย่อยได้ทุกความยาวตั้งแต่ 1 ถึง n ซึ่งการแทนตัวแปรในชั้นที่ 1 จะเป็นเซตย่อยของสายอักขระย่อยทั้งหมดที่เป็นไปได้ของสายอักขระที่กำลังพิจารณา ดังนั้นจำนวนกฎทั้งหมดที่จะสามารถแทนได้ในชั้นที่ 1 จะน้อยกว่าหรือเท่ากับ n^2 และเวลาที่ใช้การตรวจสอบกับตัวอย่างลบทุกตัวเท่ากับ $||S_N||^3$ ใช้เวลาที่แทนสายอักขระด้วยตัวแปรในชั้นแรกจะใช้เวลาไม่เกิน n^2 $||S_N||^3$ ในการตรวจสอบ ดังนั้นการแทนสายอักขระย่อยด้วยตัวแปรทั้งหมด จะใช้เวลาน้อยกว่าหรือเท่ากับ $k\, n^2\, ||S_N||^3$ เมื่อทำการพิจารณาค่า k เราจะเห็นว่า k จะเป็นจำนวนรอบที่ใช้ในการบรรยายการวนซ้ำของออโตมาตาดังนั้นจึงมีจำนวนเป็นค่าคงที่ ดังนั้นอัลกอริทึมนี้จะใช้ความซับซ้อนเชิงเวลาเท่ากับ $O(n^2\, ||S_N||^3)$

การพิจารณาอัลกอริทึมสร้างตัวแปรวนซ้ำโดยวิธีหาสายอักขระย่อยที่มีความยาวมากสุด อัลกอริทึมเริ่มจากการเปรียบเทียบสายอักขระที่แทนด้วยตัวแปร กับทุกกฎทางขวามือที่มีตัวแปรทางซ้ายมือเป็นตัวแปรเริ่มต้น ดังนั้นจำนวนรอบจะเท่ากับ จำนวนกฎที่มีตัวแปรทางซ้ายมือเป็นตัวแปรเริ่มต้นแทนด้วย m และเริ่มพิจารณาทีละคู่เพื่อหาสายอักขระย่อยที่มีความยาวมากที่สุด ให้ n แทนความยาวของสายอักขระที่แทนด้วยตัวแปร s แทนความยาวของสายอักขระของกฎทางขวามือ การหาสายอักขระย่อยที่มีความยาวมากที่สุดใช้ความซับซ้อนเชิงเวลาเท่ากับ $O((s+n)s^2)$ เพราะว่าอัลกอริทึมของคนุท มอริส แพรต ใช้ความซับซ้อนเชิงเวลาในการค้นหา O(s+n) และ เซตของสายอักขระย่อยมีจำนวนน้อยกว่าหรือเท่ากับ s^2 จากนั้นอัลกอริทึมจะทำการพิจารณารูปแบบของกฎที่จะสร้างขึ้นซึ่งแบ่งออกเป็น 5 ขั้นตอน แต่ละขั้นตอนจะใช้เวลาในการตรวจสอบกับตัวอย่างลบทุกตัวเท่ากับ $||S_N||^3$ ซึ่งแต่ละขั้นตอนจะมีรูปแบบที่เป็นไปได้ทั้งหมด เป็นจำนวนน้อยกว่า 3n ดังนั้นทั้งสามขั้นตอนจะใช้ความซับซ้อนเชิงเวลาไม่เกิน $O(n\, ||S_N||^3)$ แต่ขั้นตอนสุดท้ายเมื่อทำทั้งสามขั้นตอนเรียบร้อยแล้ว อัลกอริทึมนี้จะทำการสร้างตัวแปรเหมือนกับอัลกอริทึมการแทนสายอักขระด้วยตัวแปรดังนั้น จะใช้ความซับซ้อนเชิงเวลา

เท่ากับ $O(n^2 \ ||S_N||^3)$ เมื่อรวมทุกขั้นตอนของอัลกอริทึมนี้แล้วจะได้ว่า อัลกอริทึมนี้จะใช้ความซับซ้อนเชิงเวลาเท่ากับ $O(m\ n^2\ ||S_N||^3)$

การพิจารณาอัลกอริทึมการผสานตัวแปรใหม่ สามารถแบ่งการพิจารณาออกเป็นสองขั้นตอนคือ การเปรียบเทียบในของสายอักขระทางขวาของกฎที่สร้างขึ้นใหม่ กับการเปรียบเทียบระหว่างกฎเก่าและกฎใหม่ เมื่อพิจารณาขั้นตอนการเปรียบเทียบในสายอักขระทางขวาของกฎที่สร้างขึ้นใหม่ซึ่งจะทำการยุบรวมตัวแปรที่ละสองตัวไปเรื่อยๆ แล้วนำไปตรวจสอบกับตัวอย่างลบทั้งหมด ดังนั้น กรณีเลวร้ายที่สุดจะใช้เวลาในการคำนวณเท่ากับการหาเซตของสายอักขระย่อยที่เป็นไปได้ทั้งหมดแล้วนำไปตรวจสอบกับตัวอย่างลบทั้งหมด คือ $O(n^2\ ||S_N||^3)$ เมื่อพิจารณาขั้นตอนการเปรียบเทียบระหว่างกฎเก่าและกฎใหม่ จะเหมือนกับอัลกอริทึมสร้างตัวแปรวนซ้ำโดยวิธีหาสายอักขระย่อยที่มีความยาวมากสุด ดังนั้นจะใช้ความซับซ้อนเชิงเวลาเท่ากับ $O(m\ \ n^2\ \ ||S_N||^3)$ ดังนั้นเมื่อรวมทุกขั้นตอนของอัลกอริทึมแล้วจะได้ว่า อัลกอริทึมการผสานตัวแปรใหม่ จะใช้ความซับซ้อนเชิงเวลาเท่ากับ $O(m\ n^2\ ||S_N||^3)$

เมื่อพิจารณาความซับซ้อนแต่ละอัลกอริทึมแล้วเราจะได้ว่าเราจะใช้ความซับซ้อนเชิงเวลาของแต่ละตัวอย่างที่รับเข้ามาเท่ากับ $O(m\ n^2\ ||S_N||^3)$ เมื่อ m เป็นจำนวนกฎที่มีตัวแปรทางซ้ายมือเป็นตัวแปรเริ่มต้นที่มีอยู่ในไวยากรณ์ก่อนหน้า และ n เป็นความยาวของตัวอย่างบวกที่กำลังพิจารณา ให้ o เป็นจำนวนตัวอย่างบวกทั้งหมด และ n' เป็นความยาวของตัวอย่างบวกที่มีความยาวมากที่สุด ซึ่งจำนวนกฎทั้งหมดจะมีได้มากที่สุดเท่ากับจำนวนตัวอย่างบวกทั้งหมด ดังนั้นความซับซ้อนเชิงเวลาของอัลกอริทึมจะเท่ากับ $O(o^2\ n'^2\ ||S_N||^3)$ และเมื่อทำการพิจารณา $||S_P||$ จะเห็นว่า $O(o^2\ n'^2) = O(||S_P||^2)$ ดังนั้นความซับซ้อนเชิงเวลาของอัลกอริทึมจะเท่ากับ $O(||S_P||^2\ ||S_N||^3)$ ซึ่งอยู่ในฟังก์ชั่นพหุนาม

■

บทที่ 4

ผลการทดลอง

เนื่องจากงานวิจัยนี้ ได้นำเสนออัลกอริทึมการอนุมานไวยากรณ์ไม่พึ่งบริบท ดังนั้นการทดสอบสามารถทำได้โดยกำหนดภาษาตัวอย่างแล้วสร้างเซตของตัวอย่างบวกและตัวอย่างลบจากภาษาที่กำหนดขึ้น ซึ่งภาษาที่ใช้ในการทดสอบนำมาจากงานวิจัย [3] ซึ่งงานวิจัย [3] ได้ทดสอบกับตัวอย่างภาษาเหล่านี้ เพื่อแสดงว่าอัลกอริทึมสามารถอนุมานไวยากรณ์ไม่พึ่งบริบทได้ถูกต้อง รายการภาษาที่ใช้ในการทดสอบมีดังตารางที่ 4.1

ตารางที่ 4.1 ตัวอย่างภาษาที่ใช้ในการทดสอบการอนุมานไวยากรณ์

NO.	Decription of Language	Example	Properties
1	(aa)*	aa, aaaa, aaaaaa	Regular
2	(ab)*	ab, abab, ababab	Regular
3	$a^m b^n$ ($m \geq n \geq 1$)	aab, aaab, aaabb	Context-free
4	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

ในการทดสอบแต่ละภาษาจะใช้ตัวอย่างบวกที่มีความยาวไม่เกิน 30 ตัวอักษร และตัวอย่างลบที่มีความยาวไม่เกิน 30 ตัวอักษรเช่นกัน

ตารางที่ 4.2 ผลการทดสอบอัลกอริทึมกับตัวอย่างภาษา

NO.	Decription of Language	Result Grammar	Number of positive data used to converge
1	(aa)*	$S \rightarrow A$ $A \rightarrow Aaa \mid aa$	2
2	(ab)*	$S \rightarrow A$ $A \rightarrow Aab \mid ab$	2
3	$a^m b^n$ ($m \geq n \geq 1$)	$S \rightarrow AB$ $B \rightarrow aBb \mid b$ $A \rightarrow Aa \mid a$	3
4	Parentheses	$S \rightarrow C$ $C \rightarrow CB \mid B$ $B \rightarrow (B) \mid A$ $A \rightarrow A() \mid ()$	3
5	$\{w = w^R \mid w \in \{a,b\}^+\}$	$S \rightarrow E$ $E \rightarrow aEa \mid bEb \mid C \mid D$ $D \rightarrow bDb \mid A$ $C \rightarrow aCa \mid B$ $B \rightarrow Bb \mid b$ $A \rightarrow Aa \mid a$	5
6	Palindrome with center mark	$S \rightarrow A$ $A \rightarrow aAa \mid bAb \mid c$	2
7	Number of a's = Number of b's	$S \rightarrow E$ $E \rightarrow aEb \mid bEa \mid EC \mid ED \mid C \mid D$ $D \rightarrow bDa \mid B$ $C \rightarrow aCb \mid A$ $B \rightarrow Bba \mid ba$ $A \rightarrow Aab \mid ab$	7
8	Number of a's = 2 x Number of b's	ไม่สามารถหาได้	-

ตารางที่ 4.2 ผลการทดสอบอัลกอริทึมกับตัวอย่างภาษา (ต่อ)

NO.	Decription of Language	Result Grammar	Number of positive data used to converge
9	$\{a^i b^j c^k \mid i = j \text{ or } j = k, i,j,k > 0\}$	$S \rightarrow AD \mid CB$ $D \rightarrow bDc \mid bc$ $C \rightarrow aCb \mid ab$ $B \rightarrow Bc \mid c$ $A \rightarrow Aa \mid a$	5

จากตารางที่ 4.2 แสดงให้เห็นว่าอัลกอริทึมสามารถอนุมานไวยากรณ์ไม่พึ่งบริบทได้บางส่วนตามข้อมูลตัวอย่าง ซึ่งแต่ละภาษาที่สามารถอนุมานได้ถูกต้องและใช้ตัวอย่างบวกในการลู่เข้าจำนวนน้อย ภาษาที่ไม่สามารถทำได้ คือ ภาษาที่ 8 ผลที่ได้ออกมาอัลกอริทึม คือ ตัวแปรวนซ้ำจะซ้อนกันเป็นชั้นๆไปเรื่อยๆ โดยที่ไม่รวมกัน เนื่องจากไวยากรณ์มีความกำกวมมาก

จากตารางที่ 4.3 ผลที่ได้จากวิจัยงาน [3] สามารถทำภาษาทดสอบได้เกือบทั้งหมดแต่ภาษาที่ 9 ไม่สามารถแสดงผลได้เนื่องจากใช้เวลามาก และผลที่ได้จากอัลกอริทึมเมื่อเปรียบเทียบการใช้จำนวนตัวอย่างบวกที่ใช้ในการลู่เข้าไม่แตกต่างกันมาก

จากการวิเคราะห์การทำงานของอัลกอริทึมจะทำการหารูปแบบที่เป็นการวนซ้ำที่ตำแหน่งเดียวกัน ($A \rightarrow \alpha_1 A\gamma_1 \mid \alpha_2 A\gamma_2 \mid \beta$) ได้เพียงแค่สองรูปแบบเท่านั้น ซึ่งภาษาที่ 8 จะมีรูปแบบการวนซ้ำที่ตำแหน่งเดียวกันมากกว่าสองรูปแบบ ($A \rightarrow \alpha_1 A\gamma_1 \mid \alpha_2 A\gamma_2 \mid \alpha_3 A\gamma_3 \mid \beta$) ดังนั้นจึงทำให้การแทนตัวแปรจะไม่พบรูปแบบที่ซ้ำซ้อนมากกว่าได้ ดังนั้นจึงทำให้อัลกอริทึมไม่สามารถอนุมานภาษาที่มีรูปแบบของการวนซ้ำที่ตำแหน่งเดียวกันเกินกว่าสองรูปแบบได้

ดังนั้นจากการวิเคราะห์อัลกอริทึมและผลการทดลองแสดงให้เห็นว่าถ้าในกรณีที่ไวยากรณ์ไม่พึ่งบริบทมีจำนวนตัวแปรที่มากแต่มีความกำกวมไม่มากอัลกอริทึมในงานวิจัยนี้จะสามารถทำงานได้ดีกว่างานวิจัย [3]

ส่วนภาษาที่ไม่ได้อยู่ในระดับไวยากรณ์ไม่พึ่งบริบท ผลที่ได้จากอัลกอริทึม คือ ไวยากรณ์จะเพิ่มจำนวนขึ้นเรื่อยๆ ซึ่งสามารถระบุได้บางส่วนของภาษา หรือได้ไวยากรณ์ที่สามารถระบุได้แค่ตัวอย่างบวกที่ให้มาเท่านั้น

ตารางที่ 4.3 ผลการทดสอบของงานวิจัย [3] กับตัวอย่างภาษา

NO.	Decription of Language	Result Grammar	Number of positive data used to converge
1	(aa)*	$S \rightarrow aa \mid SS$	2
2	(ab)*	$S \rightarrow ab \mid SS$	2
3	$a^m b^n$ ($m \geq n \geq 1$)	$S \rightarrow ab \mid aS \mid Cb$, $C \rightarrow aS$	3
4	Parentheses	$S \rightarrow () \mid C) \mid SS$ $C \rightarrow (S$	3
5	$\{w = w^R \mid w \in \{a,b\}^+\}$	$S \rightarrow aa \mid bb \mid bD \mid Ca$ $C \rightarrow aa \mid ab \mid aS$ $D \rightarrow ab \mid bb \mid Sb$	6
6	Palindrome with center mark	$S \rightarrow Ca \mid bD$ $C \rightarrow ac \mid aS$ $D \rightarrow cb \mid Sb$	2
7	Number of a's = Number of b's	$S \rightarrow ab \mid ba \mid bC \mid Cb \mid SS$ $C \rightarrow aS \mid Sa$	7
8	Number of a's = 2 x Number of b's	$S \rightarrow bC \mid Cb \mid SS$ $C \rightarrow ab \mid ba \mid bD \mid Db$ $D \rightarrow aS \mid Sa$	18
9	$\{a^i b^j c^k \mid i = j \text{ or } j = k, i,j,k > 0\}$	ไม่สามารถหาได้	-

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ในงานวิจัยนี้ผู้วิจัยได้นำเสนออัลกอริทึมในการอนุมานไวยากรณ์ไม่พึ่งบริบทจากตัวอย่างบวกและตัวอย่างลบจำนวนจำกัด โดยกำหนดความสำเร็จของการเรียนรู้ด้วยหลักการจำแนกภายในจำกัดเพื่อให้ได้มาซึ่งไวยากรณ์ไม่พึ่งบริบท ซึ่งข้อกำหนดสำคัญคือ ไวยากรณ์ที่ได้ต้องไม่สามารถแจงส่วนตัวอย่างลบได้เลยแม้แต่ตัวอย่างเดียว แต่ต้องสามารถแจงส่วนตัวอย่างบวกได้ทั้งหมด จากทฤษฎีที่ได้มีการศึกษาวิจัยแล้วว่าการอนุมานไวยากรณ์ไม่พึ่งบริบทโดยอาศัยตัวอย่างบวกและตัวอย่างลบไม่สามารถทำการจำแนกภายในจำกัดเชิงเวลาและเชิงข้อมูลในฟังก์ชันพหุนามได้ แต่จากทฤษฎีที่เรานำเสนอนั้น เห็นได้ว่าอัลกอริทึมที่นำเสนอมีประสิทธิภาพโดยที่สามารถอนุมานไวยากรณ์ไม่พึ่งบริบทเชิงเวลาในฟังชันพหุนาม $O(||S_P||^2||S_N||^3)$ โดยที่ $||S_P||$ เป็นผลรวมของความยาวสายอักขระทุกตัวที่เป็นสมาชิกของเซตตัวอย่างบวก และ $||S_N||$ เป็นผลรวมของความยาวสายอักขระทุกตัวที่เป็นสมาชิกของเซตตัวอย่างลบ และ ใช้จำนวนตัวอย่างบวกจำนวนน้อยในการลู่เข้า แต่ข้อจำกัดของอัลกอริทึม คือ สามารถอนุมานภาษาในระดับไวยากรณ์ไม่พึ่งบริบทที่มีตัวแปรวนซ้ำเกิดขึ้นแบบเชิงกำหนดเท่านั้น ซึ่งภาษาสม่ำเสมอทุกภาษาสามารถสร้างไวยากรณ์ที่มีตัวแปรวนซ้ำเกิดขึ้นแบบเชิงกำหนดได้ ดังนั้นภาษาสม่ำเสมอทุกภาษาสามารถจำแนกโดยจำกัดจากอัลกอริทึมที่นำเสนอนี้ได้ แต่ภาษาไม่พึ่งบริบทสามารถมีตัวแปรวนซ้ำแบบเชิงไม่กำหนดได้ แต่การสร้างไวยากรณ์ไม่พึ่งบริบทในการบรรยายภาษานั้นไม่สามารถหลีกเลี่ยงได้ ทำให้อัลกอริทึมไม่สามารถจำแนกโดยจำกัดภาษาไม่พึ่งบริบทที่มีตัวแปรวนซ้ำแบบเชิงไม่กำหนดได้ ดังนั้นถ้านำอัลกอริทึมไปใช้กับไวยากรณ์ไม่พึ่งบริบทที่ไม่มีความกำกวมหรือ มีความกำกวมน้อย อัลกอริทึมจะสามารถทำงานได้มีประสิทธิภาพและมีความถูกต้องในการอนุมาน

ความซับซ้อนเชิงเวลาจากงานวิจัย [3] จะขึ้นอยู่กับจำนวนกฎ(ในรูปแบบทั่วไปของชอมสกี) ที่เป็นไปได้ในการสร้างแต่ละรอบของตัวอย่างบวกหนึ่งตัวเท่ากับ $O(|N|^{3K})$ โดยที่ $|N|$ เป็นจำนวนของตัวแปรในขณะนั้น และ K จะเป็นจำนวนกฎที่เป็นไปได้ในการสร้างขึ้นใหม่ในแต่ละรอบซึ่งแปรตามการจับคู่ของตัวแปรทั้งหมดในระบบ และเมื่อหากฎใหม่ได้จะทำการทดสอบกับตัวอย่างลบทุกตัวใช้ความซับซ้อนเชิงเวลาเท่ากับ $O(||S_N||^3)$ เช่นกัน ดังนั้นถ้าไม่จำกัดจำนวนกฎที่สร้างขึ้นได้ในแต่รอบ จะทำให้อัลกอริทึมของงานวิจัย [3] มีความซับซ้อนเชิงเวลาเพิ่มขึ้นเป็นเลขชี้กำลัง และถ้ากำหนดค่า K ต่ำจะทำให้ไม่สามารถหาไวยากรณ์ได้ถูกต้อง เมื่อเปรียบเทียบความซับซ้อนเชิงเวลากับงานวิจัย [3] ในกรณีที่ไวยากรณ์ที่ได้จากการอนุมานมีจำนวนตัวแปรน้อยจะทำให้งานวิจัย [3] ใช้ความซับซ้อนเชิงเวลาน้อยกว่า อัลกอริทึมที่นำเสนอ

เพราะ $|N|^{3K}$ จะน้อยกว่า $||S_P||^2$ แต่ในกรณีที่ไวยากรณ์ที่ได้จากการอนุมานมีจำนวนตัวแปรมีจำนวนมากขึ้น จะทำให้อัลกอริทึมจากงานวิจัย [3] ใช้ความซับซ้อนเชิงเวลามากกว่า เพราะต้องใช้ค่า K ที่มากขึ้น ทำให้อัตราการเติบโตของฟังก์ชั่น $|N|^{3K}$ จะมากกว่า $||S_P||^2$ ดังผลการทดลองของภาษา $\{a^i b^j c^k \mid i = j \text{ or } j = k, i,j,k > 0\}$

5.2 ข้อเสนอแนะ

เนื่องจากงานวิจัยนี้เป็นการเสนออัลกอริทึมในการเรียนรู้ภาษาไม่พึ่งบริบท พบว่ายังสามารถทำการวิจัยเพื่อพัฒนาให้งานวิจัยชิ้นนี้มีความสมบูรณ์มากขึ้น แยกเป็นข้อดังนี้

5.2.1 เนื่องจากขั้นตอนการทำงานของอัลกอริทึมยังไม่สามารถค้นหาตัวแปรวนซ้ำแบบเชิงไม่กำหนดได้ซึ่งอาจจะเกิดจากการขั้นตอนของการแทนสายอักขระด้วยตัวแปรทำให้ไม่สามารถค้นหาตัวแปรวนซ้ำเชิงไม่กำหนดได้ ดังนั้นควรจะเปลี่ยนวิธีการแทนสายอักขระด้วยตัวแปร หรือ เพิ่มขั้นตอนการสร้างตัวแปรวนซ้ำให้มีการตรวจสอบการสร้างตัวแปรวนซ้ำแบบเชิงไม่กำหนด ซึ่งต้องใช้เวลาการวิจัยเพื่อหาอัลกอริทึมที่มีประสิทธิภาพเชิงเวลาไม่เกินฟังก์ชั่นพหุนามด้วย จะทำให้งานวิจัยมีความสมบูรณ์มากขึ้น

5.2.2 การนำอัลกอริทึมไปใช้กับข้อมูลจริง ซึ่งข้อมูลที่ใช้ในความเป็นจริงอาจจะเป็นข้อมูลที่ขาดความสมบูรณ์ และข้อมูลอาจจะมีค่าความผิดพลาดเกิดขึ้นได้ ดังนั้นควรปรับปรุงอัลกอริทึมให้สามารถสร้างไวยากรณ์จากข้อมูลที่ไม่มีความสมบูรณ์ และใช้วิธีการเชิงสถิติเข้ามาช่วยสำหรับข้อมูลที่มีค่าความผิดพลาด

รายการอ้างอิง

[1] Gold, E.M. (1967). Language identification in the limit. *Information and Control*, 10(5) : 447-474.

[2] de la Higuera, C. (1997). Characteristic sets for polynomial grammatical inference. *Machine Learning Journal*, 27 : 125-138.

[3] Nakamura, K., & Matsumoto, M. (2005). Incremental learning of context-free grammars based on bottom-up parsing and search. *Pattern Recognition*, 38(9) : 1384–1392.

[4] John Cocke and Jacob T. Schwartz. (1970). Programming languages and their compilers: Preliminary notes. *Technical report, Courant Institute of Mathematical Sciences, New York University*.

[5] Earley, J. (1970). An efficient context-free parsing algorithm. *Communications of the Association for Computing Machinery* : 94-102.

[6] Angluin, D. (1983). Inductive inference: Theory and methods. Computing Surveys, 15(3) : 237-269.

[7] Oncina, J., Garcia, P. (1992). Identifying regular languages in polynomial time. *In Bunke,H., ed. Advances in Structural and Syntactic Pattern Recognition. Volume 5 of Series in Machine Perception and Artificial Intelligence. World Scientific* : 99–108.

[8] Pitt, L., Warmuth, M. (1993). The minimum consistent DFA problem cannot be approximated within any polynomial. *Journal of the Association for Computing Machinery*, 40 : 95–142.

[9] Kearns, M. and Valiant, L. (1989). Cryptographic Limitations on Learning Boolean Formulae and Finite Automata. *In 21st ACM Symposium on Theory of Computing.* : 433-444.

[10] Takada, Y. (1998). Grammatical inference for even linear languages based on control sets. *Information Processing Letters,* 28(4) : 193-199.

[11] de la Higuera, C., & Oncina, J. (2002). Learning deterministic linear languages. *In J., Kivinen, & R. H., Sloan, (Eds.), Proceedings of COLT 2002* : 185-200.

[12] Yokomori, T. (2003). Polynomial-time identification of very simple grammars from positive data. *Theor. Comput. Sci.,* (298) : 179-206.

[13] Sakakibara, Y. (1997). Ga-based learning of context-free grammars using tabular representations. *In Proceeedings of 16th International Conference on Machine learning(ICML-99)* : 354-360.

[14] Sakakibara, Y. (2005). Learning context-free grammars using tabular representations. *Pattern Recognition 38(9)* : 1372-1383.

[15] Langley P., Stromsten S. (2000). Learning context-free grammars with a simplicity bias. *Machine Learning : ECML 2000, LNAI 1810, Springer, Berlin* : 336-355.

[16] Eyraud, R., de la Hiuera, C., Janodet J. (2006). LARS: A learning algorithm for rewriting systems. *Machine Learning, Springer, Netherlands* : 7-31.

[17] Knuth, D.E., Morris, J.H., Pratt, V.R. (1977). Fast Pattern Matching in Strings. *SIAM J. Comput.6(2)* : 323-350.

ประวัติผู้เขียนวิทยานิพนธ์

นายวุฒิ สุนทรภัณฑ์ เกิดเมื่อวันที่ 28 เมษายน พ.ศ. 2525 จบการศึกษาระดับมัธยมศึกษาตอนปลายจากศูนย์การศึกษานอกโรงเรียนจังหวัดพระนครศรีอยุธยา เข้าศึกษาต่อในระดับปริญญาบัญฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย จนสำเร็จการศึกษาในปี พ.ศ. 2546 และศึกษาต่อในระดับปริญญามหาบัณฑิต ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย