บทที่ 2

ไวยากรณ์ไม่พึ่งบริบท และการแจงส่วนประโยค

2.1 ไวยากรณ์ไม่พึ่งบริบท (Context-Free Grammar)

ในการประมวลผลภาษาธรรมชาติ เพื่อให้คอมพิวเตอร์สามารถเข้าใจภาษาที่เราใช้ได้นั้น เราจะต้องประยุกต์หลักการและโครงสร้างทางภาษา ให้อยู่ในรูปฐานความรู้ที่คอมพิวเตอร์สามารถดึงมาใช้ได้ ดังนั้นนักภาษาศาสตร์คอมพิวเตอร์ จึงได้แปลงลักษณะรูปแบบการใช้ภาษาให้อยู่ในรูปของกฎเกณฑ์ต่างๆ เช่น กฎการผลิต (Production rule) ซึ่งจะอยู่ในรูปของ $\alpha \rightarrow \beta$ และมีความหมายว่า สัญลักษณ์ทางซ้ายมือ α สามารถจะถูกเขียนใหม่ได้โดยใช้สัญลักษณ์ทางขวามือ β N. Chomsky ได้จัดประเภทของภาษาออกเป็น 4 ระดับ คือ ภาษาที่ไม่มีข้อจำกัด(Unrestricted Languages), ภาษาพึ่งบริบท (Context-Sensitive Languages), ภาษาไม่พึ่งบริบท (Context-Free Languages) และ ภาษาที่มีกฎระเบียบ (Regular Languages) ระดับของภาษาต่างๆ เหล่านี้จะถูกอธิบายด้วยการเขียนเป็นกฎการผลิตต่างๆ การเลือกระดับของภาษาในการประมวลผลภาษาธรรมชาตินั้น ระดับของภาษาที่นิยมใช้ในการอธิบายภาษาและใช้ในการประมวลผล ก็คือระดับของภาษาไม่พึ่งบริบท เพราะเป็นระดับภาษาที่ครอบคลุมลักษณะต่างๆ ของภาษาได้มากกว่าภาษาในระดับของ ภาษาที่มีกฎระเบียบแน่นอน ในขณะเดียวกันก็มีความซับซ้อนของไวยากรณ์น้อยกว่าอีก 2 ระดับข้างบน แต่ก็เพียงพอที่จะใช้อธิบายกฎเกณฑ์ทางไวยากรณ์ของภาษาทั่วไปได้ และเวลาที่ใช้ในการประมวลผลก็ไม่นานนัก ทำให้มีความเหมาะสมในการนำมาใช้กับการประมวลผลภาษาธรรมชาติ

ไวยากรณ์ที่นำมาใช้ในการกำหนดภาษาของภาษาไม่พึ่งบริบทก็คือ ไวยากรณ์ไม่พึ่งบริบท (Context-Free Grammar) ตัวอย่างข้างล่างนี้จะเป็นรูปแบบง่ายๆของไวยากรณ์ไม่พึ่งบริบทสำหรับภาษาไทย ที่เราใช้ในการอธิบายขั้นตอนการทำงานต่างๆ

เมื่อให้ S แทน ประโยค (Sentence) NP แทน นามวลี (Noun Phrase) VP แทน กริยาวลี (Verb Phrase) PP แทน บุพบทวลี (Prepositional Phrase) det แทน คำชี้เฉพาะ (determiner) n แทน คำนาม (noun) v แทน คำกริยา (verb) และ p แทน คำบุพบท (preposition)

(1) $S \rightarrow NP\ VP$
(2) $S \rightarrow S\ PP$
(3) $NP \rightarrow n$
(4) $NP \rightarrow NP\ PP$
(5) $PP \rightarrow p\ NP$
(6) $VP \rightarrow v\ NP$
(7) $VP \rightarrow v\ PP$

รูปที่ 2.1 ไวยากรณ์ไม่พึ่งบริบท แบบง่าย ๆ สำหรับภาษาไทย

กฎแต่ละกฎของไวยากรณ์ไม่พึ่งบริบท จะอยู่ในรูปของ $\alpha \rightarrow \beta$ โดยที่สัญลักษณ์ทางซ้ายมือจะมีเพียงค่าเดียว และสามารถเขียนแทนได้ด้วยสัญลักษณ์ทางขวามือ สัญลักษณ์ที่ปรากฏอยู่ทางด้านซ้ายของกฎการผลิต จะเรียกว่า สัญลักษณ์ไม่ปลายทาง (Nonterminal symbol) เช่น S, NP, PP และ VP สัญลักษณ์ที่อยู่ทางด้านขวาและไม่เคยปรากฏอยู่ทางด้านซ้ายของกฎเลยจะถูกเรียกว่า สัญลักษณ์ปลายทาง (Terminal symbol) เช่น n, v, det และ p นอกจากนี้ S ยังถูกเรียกว่า สัญลักษณ์เริ่มต้น (Start symbol) ด้วย

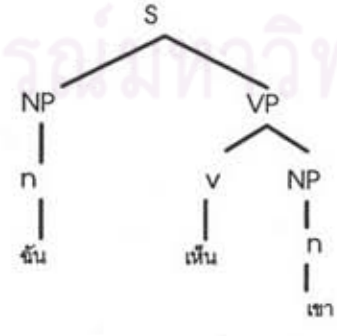
กฎการผลิต แต่ละข้อจะมีความหมายของตัวเอง เช่น ยกตัวอย่างประโยค "ผมทานข้าว"

จากกฎข้อที่ 1) S $\rightarrow$ NP VP หมายความว่า ประโยค (S) จะประกอบด้วย นามวลี (NP) ตามด้วย กริยาวลี (VP) ดังนั้นเราสามารถแยกประโยค (S) ได้เป็น "ผม" (NP) + ทานข้าว (VP) หรือ

จากกฎข้อที่ 2) S $\rightarrow$ S PP หมายความว่า ประโยค (S) อาจจะประกอบด้วย ส่วนที่เป็นประโยคย่อย (S) แล้วตามด้วย บุพบทวลี (PP) ยกตัวอย่างประโยค "ผมทานข้าวบนโต๊ะ" จะเห็นว่าเราสามารถแยกส่วนประกอบของประโยคออกเป็น "ผมทานข้าว" (S ; ตามกฎข้อ1.) + "บนโต๊ะ" (PP)
จากตัวอย่างข้างบน เราสามารถแปลความหมายของกฎต่างๆ ในไวยากรณ์ไม่พึ่งบริบทได้ในลักษณะเดียวกัน

2.2 การแจงส่วนประโยค (Parsing)

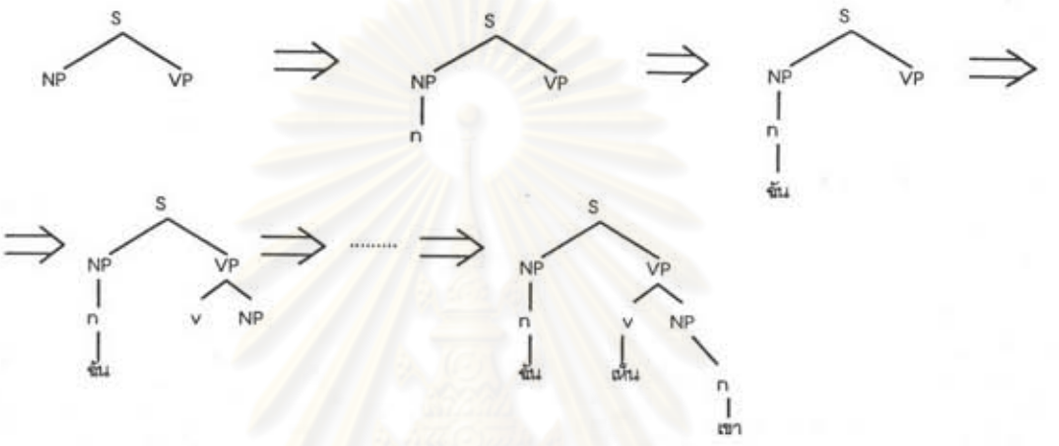
เราอาศัยกฎการผลิตต่างๆเหล่านี้ เพื่อใช้ในการตรวจวิเคราะห์โครงสร้างประโยคข้อมูลที่ต้องการ อาจพอสรุปได้ว่า การแจงส่วนประโยค ก็คือการใช้กฎเกณฑ์ไวยากรณ์ที่มี แยกแยะตรวจสอบดูว่าประโยคข้อมูลที่ต้องการ ประกอบด้วยส่วนต่างๆ อะไรบ้าง เป็นประโยคที่ถูกต้องตามไวยากรณ์หรือไม่ และมีความสัมพันธ์กันอย่างไร ซึ่งจะเป็นบันไดนำไปสู่การเข้าใจประโยคภาษาธรรมชาติ ส่วนของโปรแกรมที่ใช้ในการแจงส่วนประโยคก็จะเรียกว่า ตัวแจงส่วน (Parser) (ยืน ภู่วรวรรณ และชัยยงค์ วงศ์ชัยสุวัฒน์, 2535; Aho, 1986) และถ้าเรานำคำหรือองค์ประกอบต่างๆ ในประโยคมาเขียนความสัมพันธ์ ในรูปแบบที่เป็นโครงสร้างต้นไม้ เราก็จะเรียกการแสดงผลแบบนี้ว่าเป็น โครงสร้างต้นไม้ของการแจงส่วน (prased tree) ตัวอย่างเช่น โครงสร้างต้นไม้ของการแจงส่วนประโยค "I saw a man" จะแสดงได้ดังรูปที่ 2.2



รูปที่ 2.2 แสดงโครงสร้างต้นไม้ของประโยค "ฉัน เห็น เขา"

จากตัวอย่างที่แสดงข้างต้น จะเห็นว่าประโยคที่ต้องการแจงส่วนถูกกระจายจากสัญลักษณ์เริ่มต้น (S) เป็นองค์ประกอบย่อยที่เล็กลง พิจารณาประโยค "ฉัน เห็น เขา" การทำงานของตัวแจงส่วนโดยใช้ไวยากรณ์ไม่พึ่งบริบทในรูปที่ 2.1 อาจสรุปเป็นขั้นตอนได้ดังนี้

S : S -> NP VP -> n VP -> ฉัน VP -> ฉัน v NP
-> ฉัน เห็น NP -> ฉัน เห็น n -> ฉัน เห็น เขา

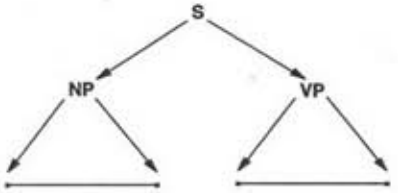


รูปที่ 2.3 แสดงขั้นตอนการวิเคราะห์ประโยค "ฉัน เห็น เขา"

การวิเคราะห์ประโยคที่แสดงข้างบนเป็นการวิเคราะห์โดยการสืบค้นจากทางซ้าย (leftmost derivation) ซึ่งเป็นการกระจายสัญลักษณ์ไม่ปลายทาง ทางด้านซ้ายของการกระจายก่อน ในทำนองเดียวกันเราก็สามารถวิเคราะห์ประโยค โดยการสืบค้นมาจากสัญลักษณ์ไม่ปลายทาง ทางด้านขวาก่อนเช่นกัน โดยจะเรียกว่า การสืบค้นจากทางขวา (rightmost derivation)

การแจงส่วนประโยคอาจกระทำได้สองวิธีคือ การแจงส่วนประโยคจากบนลงล่าง (Top-down parsing) และการแจงส่วนประโยคจากล่างขึ้นบน (Bottom-up parsing)

การแจงส่วนประโยคจากบนลงล่าง (Top-down parsing) จะเริ่มแจงส่วนประโยคจากสัญลักษณ์เริ่มต้นประโยค(S) แล้วกระจาย เป็นสัญลักษณ์ทางขวาของกฎที่ประกอบกันเป็นประโยค จากบนลงล่าง จนกว่าจะมาถึงสัญลักษณ์ปลายทาง ซึ่งเป็นคำศัพท์นั้นๆ ตัวแจงส่วนที่ใช้หลักการแจงส่วนประโยคจากบนลงล่างที่ใช้กันอยู่ในปัจจุบันก็มี เช่น ตัวแจงส่วนที่ใช้หลักการแจงส่วนแบบ Earley (Earley's parsing algorithm) (Earley, 1970; Tanaka, 1993)



รูปที่ 2.4 การแจงส่วนประโยคจากบนลงล่าง (Top-down parsing)

การแจงส่วนประโยคจากล่างขึ้นบน (Bottom-up parsing) จะเริ่มจากสัญลักษณ์ปลายทาง ซึ่งเป็นคำศัพท์ต่างๆ แล้วถูกแทนด้วยชนิดของคำ (category) ของคำศัพท์นั้นๆ จากนั้นจะแทนสัญลักษณ์ทางขวาของกฎด้วยสัญลักษณ์ทางซ้ายตามลำดับขึ้นไปจนกว่าจะพบ สัญลักษณ์เริ่มต้นประโยค(S) ตัวแจงส่วนที่ใช้หลักการแจงส่วนประโยคจากล่างขึ้นบน ที่ใช้กันอยู่ในปัจจุบันก็มี เช่น ตัวแจงส่วนที่ใช้หลักการแจงส่วนแบบแผนภูมิ (Chart parsing algorithm) (Kay, 1980) ตัวแจงส่วนแบบแอลอาร์ (LR parser) (Aho, 1986) และตัวแจงส่วนแบบจีแอลอาร์ (GLR parser) (Tanaka,1992,1993; Tomita, 1991)

รูปที่ 2.5 การแจงส่วนประโยคจากล่างขึ้นบน (Bottom-up parsing)