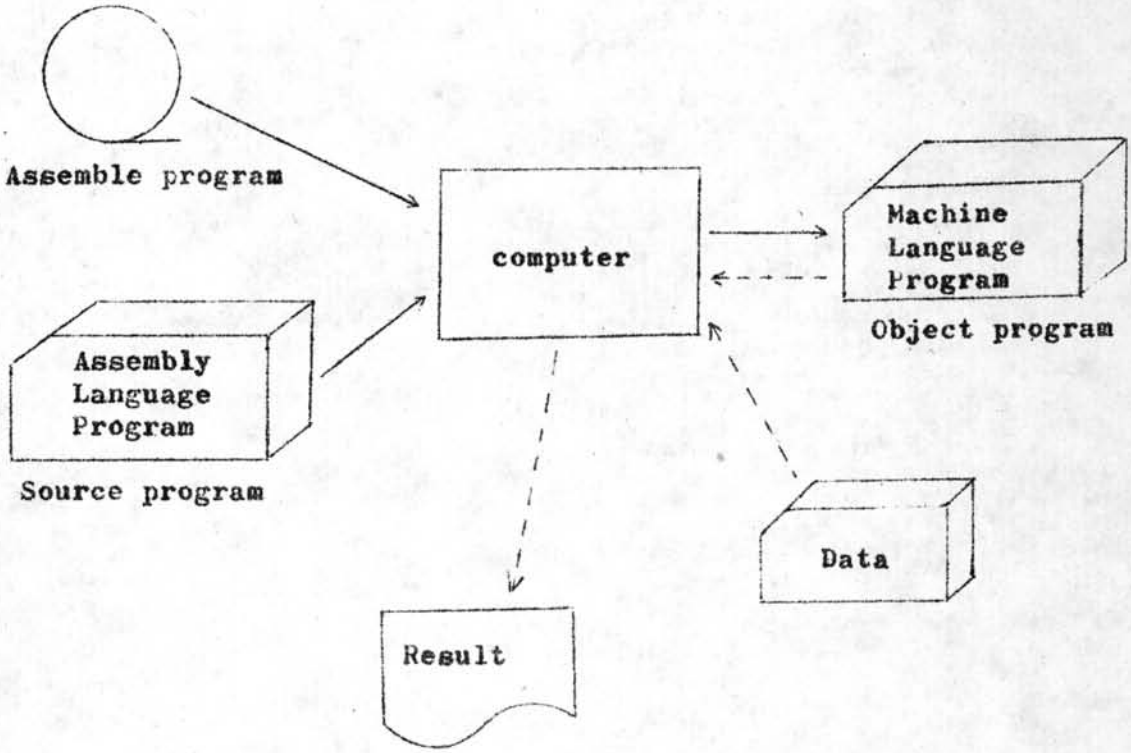
บทที่ 2

ลักษณะการแปลโปรแกรม

ในยุคแรกของการนำคอมพิวเตอร์มาใช้ การเขียนโปรแกรมให้คอมพิวเตอร์ทำงาน ผู้เขียนโปรแกรมต้องเขียนคำสั่งเป็นภาษาเครื่อง ซึ่งเป็นภาษาระหัสตัวเลขที่คอมพิวเตอร์สามารถตีความได้ การตั้งระหัสต่าง ๆ แล้วแต่ระบบการทำงานของเครื่องคอมพิวเตอร์แต่ละแบบ ดังนั้นผู้เขียนโปรแกรมจะต้องมีความเข้าใจเกี่ยวกับระบบการทำงานของเครื่องเป็นอย่างดี และเนื่องจากเป็นการสั่งงานโดยตรง จึงต้องสั่งงานให้ละเอียดทุกขั้นตอน และติดตามที่อยู่ของข้อมูลนั้น ๆ อย่างถูกต้อง ต่อมาเพื่อลดความยุ่งยากในการเขียนโปรแกรม จึงได้มีการพัฒนาโปรแกรมโดยใช้ สัญญลักษณ์แทน ระหัสตัวเลข เรียกว่า โปรแกรมภาษาแอสเซมบลี้ (Assembly language) ดังรูป 2.1 ทำให้การเขียนโปรแกรมได้สะดวกและเข้าใจได้ง่ายขึ้น ถึงขั้นนี้จึงจำเป็นที่จะต้องแปลภาษาแอสเซมบลี้ที่เขียนขึ้นนี้ให้เป็นภาษาเครื่องที่เครื่องคอมพิวเตอร์สามารถเข้าใจได้ก่อนทำงานตามคำสั่งนั้น จึงมีผู้เขียนโปรแกรมขึ้นมาเป็นพิเศษสำหรับใช้ให้เครื่องแปลจากภาษาแอสเซมบลี้เป็นภาษาเครื่อง เรียก แอสเซมบลี้โปรแกรม (Assemble program) (ดังรูป 2.2) โดยโปรแกรมที่เขียนขึ้นก่อนเรียกว่า ซอสโปรแกรม (Source program) และโปรแกรมภาษาเครื่องที่จะนำไปใช้งานเรียกออบเจคโปรแกรม (Object program)

โปรแกรมภาษาแอสเซมบลี้	โปรแกรมภาษาเครื่อง
BS 4	35 000004
LCA #2, SUM	15 010400 010430
A +1, SUM	36 010470 010430
C LAST, 4	33 010476 000004
BA #3B1, 4	34 010300 000004

รูป 2.1 เปรียบเทียบระหว่างภาษาเครื่องกับภาษาแอสเซมบลี้



รูป 2.2 โปรแกรมแปลภาษาแอสเซมบลี้

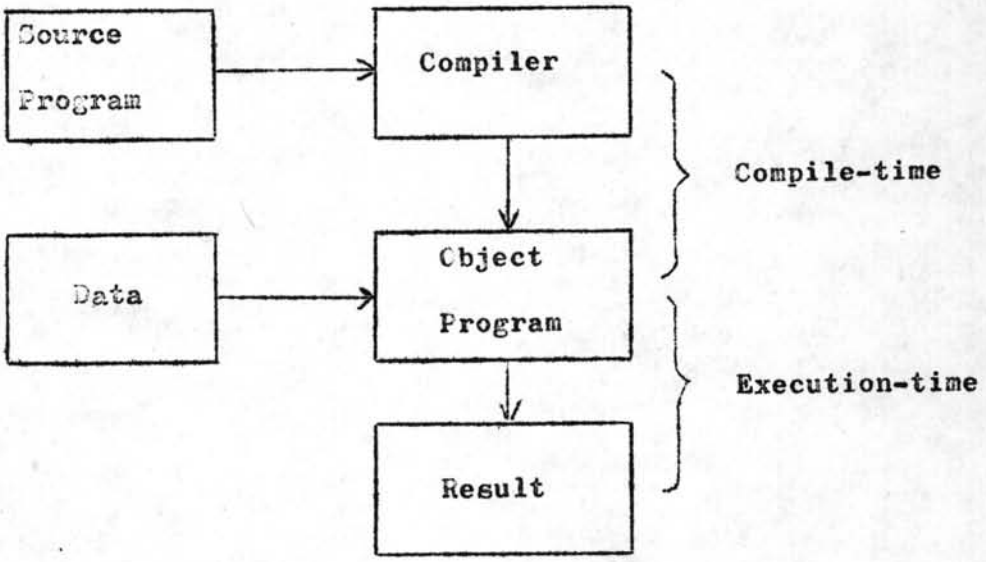
จากรูป 2.1 จะเห็นว่าภาษาแอสเซมบลี้กับภาษาเครื่องคล้ายกันมาก การเขียนคำสั่งยังคงจะต้องเขียนให้ละเอียดทุกขั้นตอน ได้มีการคิดค้นภาษาเขียนขึ้นมาใหม่ โดยให้สามารถสั่งงานได้อย่างกว้าง ๆ แล้วให้โปรแกรมแปลแตกแยกคำสั่งเหล่านี้ให้เป็นขั้นตอนโดยละเอียด ทำให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมโดยคำนึงถึงความมุ่งหมายของโปรแกรมนั้นไม่ต้องคอยพะวงเกี่ยวกับรายละเอียดขั้นตอนการทำงาน ภาษาเหล่านี้มีหลายภาษา เรียก<u>ภาษาระดับสูง</u> (High-level language) เช่น ฟอร์แทรน, โคบอล, พีแอล/1 เป็นต้น โปรแกรมพิเศษที่ใช้แปลภาษาเหล่านี้เรียก <u>ตัวแปลโปรแกรม</u> (Compiler)

<u>ลักษณะการแปลโปรแกรม</u>

ส่วนใหญ่แล้วลักษณะการเขียนตัวแปลโปรแกรมโดยทั่ว ๆ ไปจะมีหลักการคล้ายกัน ที่แตกต่างกันไปก็เกี่ยวกับการจัดระบบการทำงานของเครื่องคอมพิวเตอร์โดยทั่วไป ตัวแปลโปรแกรมมักแบ่งออกเป็นโปรแกรมย่อยหลาย ๆ ส่วน แต่ละส่วนมีจุดประสงค์ในการทำงานอย่างหนึ่ง มีข้อมูล อินพุท (Input) และเอาท์พุท (Cutput) เมื่อสิ้นสุดการทำงานของโปรแกรมแต่ละส่วน ในคอมพิวเตอร์ขนาดกลางและขนาดเล็กซึ่งมีหน่วยเก็บ (Main Storage) จำกัดสำหรับขนาดของโปรแกรม จึงเป็นไปไม่ได้ที่จะนำตัวแปลโปรแกรมทั้งหมดเข้าไปเก็บเลยทีเดียว ดังนั้นในการทำงานแปลโปรแกรม โปรแกรมคอมพายเลอร์ซึ่งถูกแบ่งออกเป็นส่วน ๆ แล้ว จะถูกเรียกเข้าไปเก็บในหน่วยความจำเพื่อใช้งานที่จะทำในขณะนั้น เฉพาะโปรแกรมที่ต้องใช้จริง ๆ เมื่องานนั้นแล้วเสร็จ ส่วนของโปรแกรมที่จะทำงานต่อไป จึงจะถูกเรียกเข้าไปใช้ เป็นเช่นนี้เรื่อย ๆ จนเสร็จสิ้นงานแปล

ในคอมพิวเตอร์ที่มีหน่วยความจำขนาดใหญ่ สามารถที่จะเก็บตัวแปลโปรแกรมทั้งหมดเข้าไปใช้งานในครั้งเดียว เรียก คอมพลีทคอมพายเลอร์ (Complete Compiler) ลักษณะการทำงานแบบนี้จะใช้เวลาในการแปลโปรแกรมน้อยกว่าในคอมพิวเตอร์ที่มีหน่วยความจำขนาดกลางและขนาดเล็ก อย่างน้อยที่สุดก็ประหยัดเวลาในการเรียกโปรแกรมย่อยของตัวแปลโปรแกรมแต่ละส่วน (I/Ø-Time) แต่ทั้งนี้ ก็ขึ้นอยู่กับความเหมาะสมของการใช้คอมพิวเตอร์ ซึ่งคอมพิวเตอร์ที่จะทำงานแบบคอมพลีท-คอมพายเลอร์จะต้องเป็นเครื่องที่มีหน่วยเก็บ (Main Storage) ที่ใหญ่มาก ค่าใช้จ่ายอื่น ๆ ก็ย่อมจะต้องมากขึ้นด้วย

ในรูป 2.3 แสดงให้เห็นตัวแปลโปรแกรม จะแปลซอสโปรแกรม (Source Program) ให้เป็นออบเจคโปรแกรม (Object-program) ก่อนในช่วงเวลาที่เรียกคอมพายไทม์ (Compile-Time) แล้วจึงรับข้อมูล (Data) เข้าไปคำนวณผลที่ต้องการ



รูป 2.3 การคำนวณโปรแกรมภาษาระดับสูง (High-level languge)

โครงสร้างทั่วไปของตัวแปลโปรแกรม (Structure of Compiler)

โดยทั่วไปลักษณะโครงสร้างของตัวแปลโปรแกรม ประกอบด้วยส่วนที่ทำการวิเคราะห์ (Analysis), ส่วนที่ทำงานสังเคราะห์ (Synthesis) และส่วนที่ทำหน้าที่ในการจัดการเกี่ยวกับลักษณะและชนิดของข้อมูล ให้เก็บไว้ในตารางสำหรับเก็บข้อมูลหรือข่าวสารชนิดนั้น ๆ (Tables of information) เพื่อใช้งานในแต่ละขั้นตอนของการแปล ในแต่ละตารางจะมีข้อมูลซึ่งใช้งานในลักษณะเดียวกัน เช่น ตารางสำหรับเก็บชื่อตัวแปร (Symbol table or name table) ก็จะเก็บชื่อหรือตัวแปรที่เขียนในซอสโปรแกรม (Source Program) พร้อมทั้งจัดลักษณะ, ชนิด ที่อยู่ (Address) และรายละเอียดอื่น ๆ ให้, ตารางเก็บค่าคงที่ (constant tables), ตารางแสดงความสัมพันธ์ของคำสั่งและตารางอื่น ๆ อีกมาก ลักษณะของตารางเหล่านี้จะเปลี่ยนแปลงเพิ่มเติมหรือถูกลบไปเมื่อใช้งานแล้ว ทั้งนี้ ขึ้นอยู่กับขั้นตอนต่าง ๆ ในการทำงานระหว่างการแปล รูป 2.4 แสดงถึงโครงสร้างของตัวแปลโปรแกรม

ส่วนทำการวิเคราะห์ซอสโปรแกรม

เมื่อเริ่มต้นการแปลซอสโปรแกรม ข้อมูลแต่ละกระทงความ (Statement) ถูกอ่านเข้ามาเมื่อตรวจคอลัมน์แรกแล้วไม่ใช่ข้อความอธิบาย ก็จะกำจัดส่วนที่ไม่เกี่ยวในคำสั่ง เช่น ช่องว่าง ซึ่งระหว่างเขียนโปรแกรมอาจเว้นไว้เพื่อความเป็นระเบียบหรือดูง่าย ดังนั้น ข้อมูลในคำสั่งจะถูกเลื่อนชิดติดกันหมดแล้วจึงเริ่มสแกนคำสั่งโดยตัวแปลโปรแกรมส่วนที่เป็นสแกนเนอร์ (Scanner) โดยจะทำการสแกนจากซ้ายไปขวา เมื่อทราบตัวใดเป็นคำสั่ง, ตัวแปร, ตัวคงที่ ฯลฯ ก็จะจัดส่งไปยังตารางเก็บข่าวสาร (Tables of information) ในขณะเดียวกันมีการสร้างรูปแบบภายในของซอสโปรแกรม (Internal form of source program) เก็บไว้ในพื้นที่ส่วนหนึ่งของหน่วยความจำที่เตรียมไว้ ซึ่งโปรแกรมภายในนี้จะเป็นตัวแทนของซอสโปรแกรมในการแปลขั้นตอนต่อไปในระหว่างการสแกนก็จะตรวจสอบความผิดพลาดของซอสโปรแกรม

การแปลความหมายของคำสั่งและตรวจสอบความผิดพลาดเหล่านี้ จะกระทำโดย ตัวแปลโปรแกรมส่วนที่เรียกการวิเคราะห์ตรวจสอบและแปลความหมาย (Syntax and Semantic Analyzers) สิ่งที่ได้หลังจากเสร็จสิ้นการทำงานส่วนนี้ ก็คือ โปรแกรมภายในที่เป็นตัวแทนของซอสโปรแกรม และตารางต่าง ๆ ที่เก็บสัญญลักษณ์, ตัวคงที่, ความสัมพันธ์ในคำสั่ง เพื่อใช้สร้างออบเจคโปรแกรมต่อไป

<u>ตัวอย่าง</u> คำสั่ง A = B + C * D

จะถูกแยกเป็น * C, D, T1

+, B, T1, T2

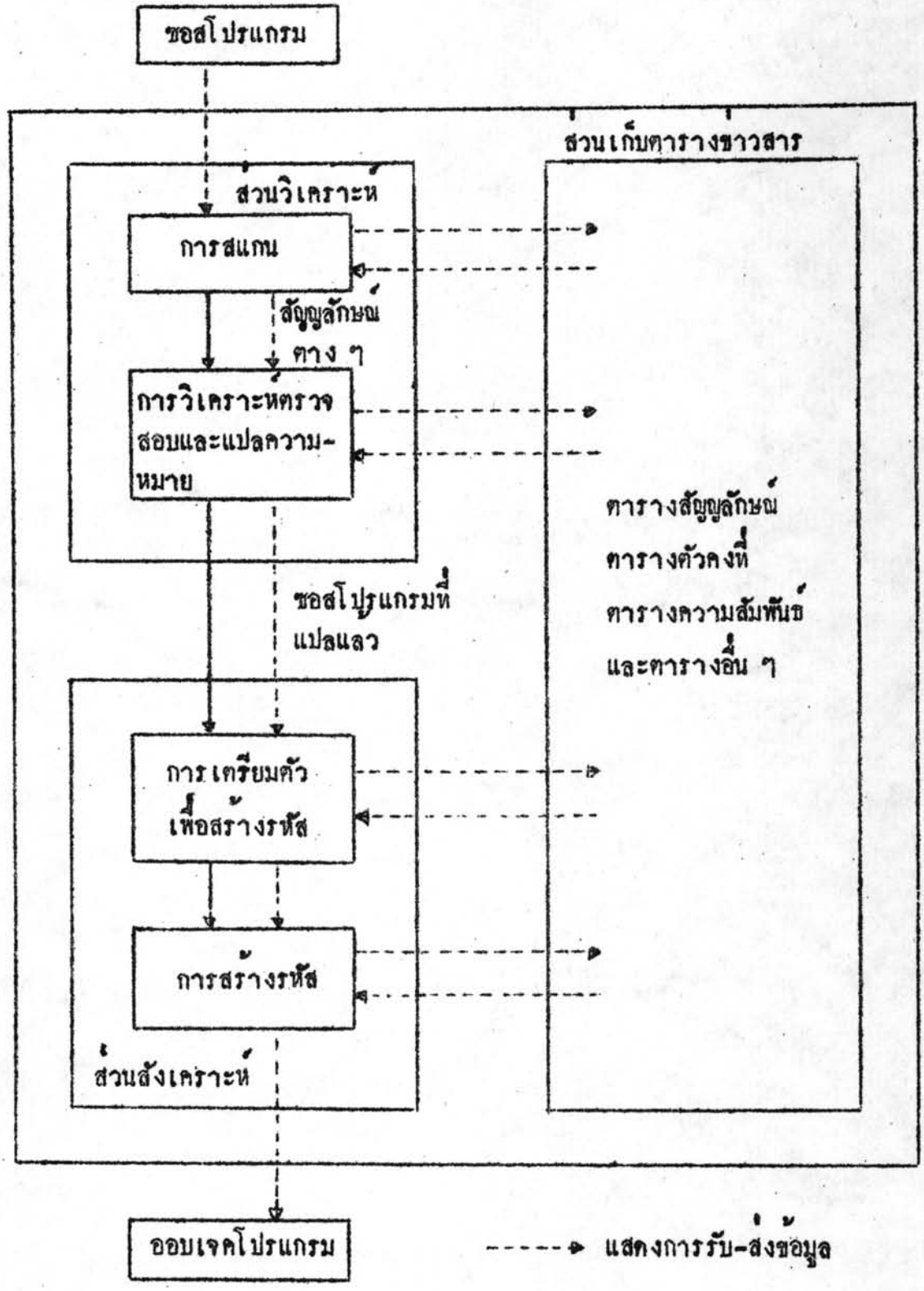
=, T2, A

T1 และ T2 เป็นตัวแปรที่ตัวแปลโปรแกรมสร้างขึ้น การแยกคำสั่งออกมาให้เป็นแต่ละขั้นตอนการทำตามคำสั่งอาศัยหลักเกณฑ์ทางคณิตศาสตร์ ตัวโอเปอแรนด์ (operands) และโอเปอเรเตอร์ (operators) ต่าง ๆ จะถูกเก็บไว้ในตาราง สำหรับเก็บโอเปอแรนด์ หรือโอเปอเรเตอร์ โดยมีดัชนี (index) แสดงความสัมพันธ์

<u>การเตรียมการและการสร้างรหัส</u>

(preparation and code generation)

ก่อนที่จะทำการสร้างระหัสให้เป็นออบเจคโปรแกรม จะต้องเปลี่ยนแปลงบางส่วนของโปรแกรมภายในที่ได้สร้างไว้ เพื่อให้สอดคล้องกับวัตถุประสงค์ของซอสโปรแกรมในฟอร์แทรน คำสั่ง จำพวก COMMON, EQUIVALENCE จะถูกพิจารณา จากนั้นตัวแปลโปรแกรมจะทำการออบติไมซ์ (optimization) เพื่อให้ออบเจคโปรแกรมที่จะถูกสร้างขึ้นใช้เวลาการคำนวณ (Execution time) น้อยที่สุด เมื่อพร้อมแล้ว โปรแกรมภายในก็จะถูกเปลี่ยนให้เป็นภาษาเครื่องหรือภาษาแอสเซมบลี้ซึ่งเป็นออบเจคโปรแกรมที่จะใช้กับเครื่องคอมพิวเตอร์นั้นต่อไป



รูป 2.4 แสดงโครงสร้างของตัวแปลโปรแกรม.