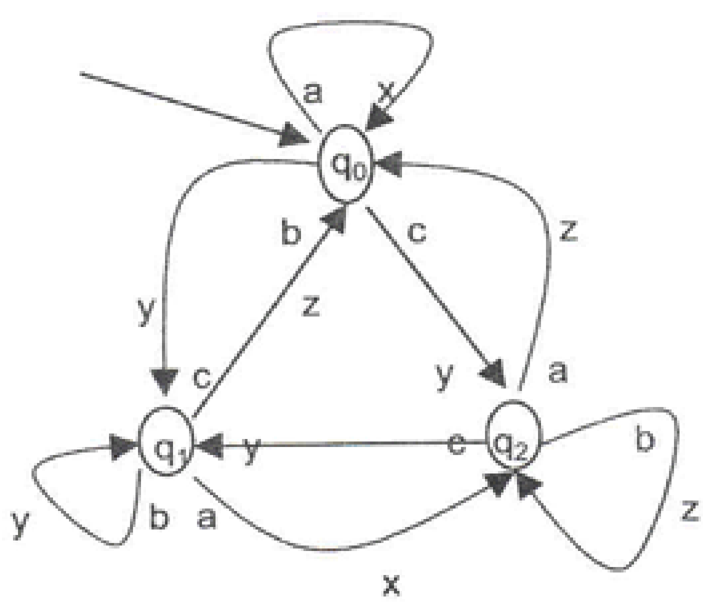
เครื่องสถานะจำกัด

(Finite State Machines)

นาวาเอก ผู้ช่วยศาสตราจารย์ ปรีดี จุลสำลี

ท่านทราบไหมว่า ก่อนที่จะมีการสร้างเครื่องคอมพิวเตอร์นั้นมีตัวแบบหนึ่งในเชิงคณิตศาสตร์ที่เป็นตัวดลใจหรือเป็นตัวชักนำทำให้เกิดเครื่องคอมพิวเตอร์ขึ้น ตัวแบบที่ว่านี้ใช้ได้จริง ๆ ก่อนการผลิตเครื่องคอมพิวเตอร์ที่คนส่วนใหญ่ใช้กันในปัจจุบัน ถ้าเราเข้าใจตัวแบบเชิงคณิตศาสตร์นี้ ก็จะช่วยทำให้เราเข้าใจว่าการคำนวณเข้าไปอยู่ในโปรแกรมที่ทำงานในเครื่องคอมพิวเตอร์ได้อย่างไร ตัวแบบที่จะกล่าวนี้เป็นเครื่องมือหนึ่งจากหลาย ๆ เครื่องมือ ที่แสดงภาวะการคำนวณได้ ตัวแบบเชิงคณิตศาสตร์นี้คือ เครื่องสถานะจำกัด (Finite State Machines) ก่อนที่จะรู้จักเครื่องชนิดนี้ เราต้องมีความรู้ด้านคณิตศาสตร์และกราฟมาก่อนบ้าง เช่น ความรู้ในเรื่องของ เซต ฟังก์ชัน กราฟ เป็นต้น เราจะนำข้อมูลต่าง ๆ มาเขียนแสดงเป็นรูปกราฟระบุทิศทางพร้อมกับมีข้อมูลต่าง ๆ อยู่บนกราฟนั้นจะทำให้เห็นและเข้าใจในตัวแบบของการคำนวณ จากนั้นก็นำตัวแบบไปสร้างเป็นฟังก์ชัน และรายละเอียดต่าง ๆ ได้ หรือในทางกลับกันถ้าเรารู้ฟังก์ชันและรายละเอียดข้อมูลต่าง ๆ เราก็นำมาเขียนเป็นกราฟระบุทิศทางได้เหมือนกัน กราฟระบุทิศทาง ที่แสดงการผ่านข้อมูลนี้เราเรียกว่า State Diagram หรือ Transition Diagram หรือเป็นเครื่องสถานะจำกัด เครื่องหนึ่งนั่นเอง ก่อนอื่นเรามาทำความเข้าใจ เกี่ยวกับเครื่องนี้ก่อน เช่น



State Diagram

จะเห็นว่า มี node อยู่ ๓ nodes ได้แก่ q_0 , q_1 และ q_2 และให้ ข้อมูลที่อยู่ฐานลูกศร เป็นข้อมูลเข้า (input data) และข้อมูลที่อยู่ตอนปลายลูกศรเป็นข้อมูลออกหรือผลลัพธ์ (output data) บนแต่ละ node อาจมีข้อมูลเข้า มากน้อยแค่ไหนขึ้นอยู่กับผู้ออกแบบสร้างและต้องเป็นจำนวนจำกัด และข้อมูลออกก็ต้องมีด้วยเช่นกัน ในที่นี้แต่ละ node มีข้อมูลเข้า เป็นตัวอักขระ a,b,c และผลลัพธ์เป็น x,y,z ถ้า node ใดมีลูกศรเข้า node ไม่มีข้อมูลกำกับในที่นี้ ได้แก่ q_0 ถือว่าเป็น node เริ่มต้นและจะมี node เดียวเท่านั้นของแต่ละเครื่อง การเริ่มต้นต้องเริ่มที่ node นี้เสมอ นี้คือ เครื่องสถานะจำกัดในรูปของ state Diagram

เมื่อมาพิจารณาการทำงานของเครื่องนี้ สมมติว่าเรามีข้อมูลเข้าเป็นสายอักขระ (string) $w = abaccbccaba$ จะได้ผลลัพธ์ (output data) เป็นอะไร ?

เริ่มการพิจารณา อักขระตัวแรกได้แก่ a ใส่เข้าไปที่ node สถานะเริ่มต้น q_0 ได้ผลลัพธ์เป็น x ตอนนี้ยังอยู่ที่สถานะ q_0 ใส่ข้อมูลเข้าตัวต่อไปคือ b ได้ผลลัพธ์เป็น y เครื่องย้ายไปอยู่ที่สถานะ q_1 ใส่ข้อมูลเข้าตัวต่อไปคือ a ได้ผลลัพธ์เป็น x เครื่องเปลี่ยนไปอยู่ที่สถานะ q_2 ต่อไปใส่ข้อมูลเข้า c ได้ผลลัพธ์เป็น y เครื่องกลับไปอยู่ที่สถานะ q_1 ใส่ข้อมูลเข้าเป็น c ได้ผลลัพธ์เป็น z เครื่องเปลี่ยนไปอยู่ที่สถานะ q_0 ใส่ข้อมูลเข้าต่อไปเรื่อยๆ จนหมดสายอักขระเข้า จะได้ผลลัพธ์ทั้งหมดคือ "$xyxyzyzyzyx$" หรือเขียนเป็นลำดับได้ดังนี้

$$q_0 \xrightarrow[x]{a} q_0 \xrightarrow[y]{b} q_1 \xrightarrow[x]{a} q_2 \xrightarrow[y]{c} q_1 \xrightarrow[z]{c} q_0$$

$$\xrightarrow[y]{b} q_1 \xrightarrow[z]{c} q_0 \xrightarrow[y]{c} q_2 \xrightarrow[z]{a} q_0 \xrightarrow[y]{b} q_1 \xrightarrow[x]{a} q_2$$

อักขระที่อยู่ข้างบนลูกศรเป็นข้อมูลเข้าและอักขระที่อยู่ข้างล่างลูกศรเป็นผลลัพธ์และ node q_0, q_1 และ q_2 เป็นสถานะ (state)

ถ้าพิจารณาข้อมูลทั้งหมดที่อยู่ใน state diagram ข้างบน เราจะรู้ว่า มีข้อมูลอยู่ ๕ ชนิดด้วยกันกล่าวคือ

๑. ข้อมูลเข้า เรียกว่า input symbol ในที่นี้มีสมาชิกเป็น a,b,c ให้เป็นเซตจำกัดเซตหนึ่ง เขียนแทนด้วย $\sum$

๒. ข้อมูลที่ผลลัพธ์เรียกว่า output symbol ในที่นี้มีสมาชิกเป็น x,y,z ให้เป็นเซตจำกัดเซตหนึ่ง เขียนแทนด้วย Z

๓. ข้อมูลที่เป็นสถานะ เรียกว่า internal state ในที่นี้มีสมาชิกเป็น q_0, q_1, q_2 ให้เป็นเซตจำกัดเซตหนึ่ง เขียนแทนด้วย S

๔. ตัวเปลี่ยนสถานะ เรียกว่า ฟังก์ชั่นการผ่าน (next state function) ในที่นี้เราสามารถนิยามเป็น

$$f(q_0,a)=q_0 \quad , \quad f(q_1,a)=q_2 \quad , \quad f(q_2,a)=q_0$$
$$f(q_0,b)=q_1 \quad , \quad f(q_1,b)=q_1 \quad , \quad f(q_2,b)=q_2$$
$$f(q_0,c)=q_2 \quad , \quad f(q_1,c)=q_0 \quad , \quad f(q_2,c)=q_1$$

ก็คือ ฟังก์ชัน $f: S\times\sum \to S$

๕. ตัวเปลี่ยนสถานะผลลัพธ์ เรียกว่า ฟังก์ชั่นผลลัพธ์ (output function) ในที่นี้เราสามารถนิยามเป็น

$$g(q_0,a)=x \quad , \quad g(q_1,a)=x \quad , \quad g(q_2,a)=z$$
$$g(q_0,b)=y \quad , \quad g(q_1,b)=y \quad , \quad g(q_2,b)=z$$
$$g(q_0,c)=y \quad , \quad g(q_1,c)=z \quad , \quad g(q_2,c)=y$$

ก็คือ ฟังก์ชัน $g: S\times\sum \to Z$

เมื่อถึงตอนนี้ก็จะทราบส่วนประกอบต่าง ๆ ของเครื่องสถานะจำกัดแล้ว และเห็นว่าจากการออกแบบสร้างเครื่องสถานะจำกัดเครื่องหนึ่งขึ้นมา ก็สามารถเอาข้อมูลต่าง ๆ ที่สร้างขึ้นนำมาแปลงเป็นฟังก์ชันได้ และก็สามารถหาคำตอบจากเครื่องดังกล่าวได้ถ้าเราใส่ข้อมูลลงไป คำตอบที่ได้จะเป็นไปตามระบบของเครื่องนั้น

สังเกตว่า ถ้าต้องการออกแบบเครื่องสถานะจำกัดเครื่องหนึ่งต้องรู้ส่วนประกอบ ๓ ส่วนด้วยกันก่อน กล่าวคือ ส่วนที่เป็น input symbol output symbol และ internal state แล้วนำไปสร้างฟังก์ชั่นการผ่าน และฟังก์ชั่นผลลัพธ์ต่อไป

ลองมาออกแบบสร้างเครื่องสถานะจำกัด เพื่อคำนวณการบวกเลขฐานสองโดยให้เครื่องนี้สามารถคำนวณได้ทุกนิพจน์ ของเลขฐานสอง ในการออกแบบสร้างเครื่องนี้ต้องทราบวิธีการบวกของระบบเลขฐานสองก่อน จากนั้นก็หาเซตของ input symbol เซตของ output symbol และเซตของสถานะภายใน (internal state) ของเครื่องนี้ แล้วนำไปเขียนเป็น State Diagram และ ฟังก์ชันการผ่านต่อไป

วิธีการ สมมติว่าต้องการบวกเลขฐานสอง 1100101101 + 1001110111

```
  1100101101
              +
  1001110111
  ----------
  0110100100
  ----------
```

๑. **หาเซตของ** input symbol

เมื่อมาพิจารณาการบวกกันตามตัวอย่างนี้สามารถนำเอาข้อมูลซึ่งเป็นสัญลักษณ์ตัวเลขฐานสองของการบวกกันจับเป็นคู่ ๆ จากขวาไปซ้าย ได้ดังนี้

11, 01, 11, 10, 01, 11, 01, 00, 10, 11

จะเห็นว่าข้อมูลทั้งหมดนี้เป็น input symbol คือเป็นสัญลักษณ์เข้า และคู่ลำดับที่เป็นสัญลักษณ์เหล่านี้มีอยู่ ๔ คู่ เท่านั้นคือ 00, 01,10 และ 11 เมื่อมาพิจารณาถึงโอกาศของคู่ลำดับในการบวกกันของทุกนิพจน์ของเลขฐานสองนั้นที่เป็นไปได้ของ input symbol จะได้เป็นเซต

$$\sum = \{00,01,10,11,bb\}$$

bb หมายถึง โอกาสที่เป็นสัญลักษณ์ว่างบวกกับสัญลักษณ์ว่าง (blank + blank)

๒. **หาเซตของ** output symbol

จากการบวกกันตามตัวอย่างจะได้ผลลัพธ์จากขวาไปซ้ายดังนี้

0, 0, 1, 0, 0,1, 0,1,1, 0,1

และข้อมูลเหล่านี้เป็น output symbol คือเป็นสัญลักษณ์ออก เห็นว่ามีสัญลักษณ์อยู่ ๒ ตัวเท่านั้น คือ 0 และ 1 เมื่อมาพิจารณาถึงโอกาสของสัญลักษณ์ออกของทุกนิพจน์ของเลขฐานสอง ได้เป็นเซต

$$Z = \{0,1,b\}$$

b หมายถึง โอกาสที่ได้ผลลัพธ์เป็นสัญลักษณ์ว่าง

๓. **หาเซตของ สถานะภายใน** (internal state)

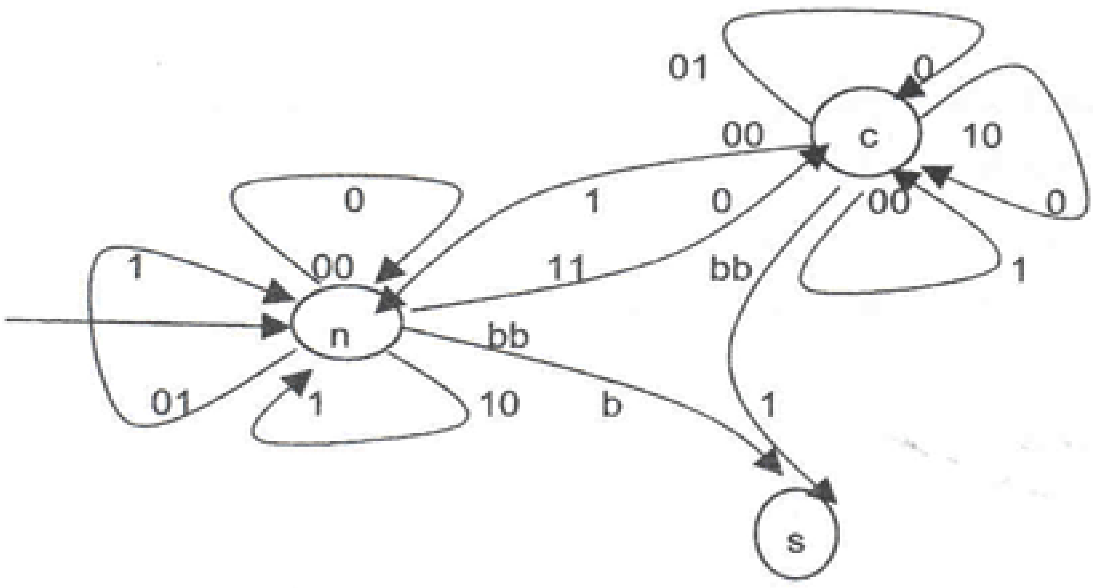
จากการบวกเลขฐานสอง รู้แล้วว่าวิธีบวกนั้นเมื่อบวกกันแล้วผลลัพธ์มีเศษกับไม่มีเศษนั่นคือเป็นสถานะภายในของการคำนวณ จึงจัดให้เป็น ๒ nodes และจำเป็นต้องมีสถานะหยุดอีกสถานะหนึ่ง ดังนั้นจึงมี ๓ สถานะด้วยกันดังนี้

n เป็นสถานะของการบวกกันแล้วไม่มีเศษ (not carry)

c เป็นสถานะของการบวกกันแล้วเก็บเศษ (carry)

s เป็นสถานะสิ้นสุดการคำนวณ (stop)

ถึงตอนนี้จึงนำมาสร้างเป็น แผนภาพการผ่าน (state diagram) ได้ดังนี้



State Diagram

วิธีการสร้าง แผนภาพการผ่านก็ไม่ยาก ที่ node n ถ้าผลการบวกกันของเลขสองตัวไม่มีเศษก็ให้อยู่ที่ n นี้ แต่ถ้าผลการบวกกันของเลขสองตัวมีเศษให้ข้ามไปอยู่ที่ node c ที่ c ถ้าผลการบวกกันของเลขสองตัวมีเศษก็ให้อยู่ที่ c นี้ แต่ถ้าผลการบวกกันไม่มีเศษให้ข้ามไปอยู่ที่ n (อย่าลืมว่าที่ c ในตัวมันเองนั้นเก็บเศษอยู่ ๑ แล้ว)

มาทดสอบการบวกกันตามโจทย์ตัวอย่างข้างต้นคือ 1100101101 + 1001110111 ด้วยเครื่องสถานะจำกัดแสดงด้วย state diagram ที่ออกแบบสร้างขึ้น มีคู่ลำดับที่เป็นข้อมูลเข้าคือ 11, 01, 11, 10, 01, 11, 01, 00, 10, 11 และ bb เริ่มต้นที่ n ดูตาม state diagram ข้างบน จะได้ลำดับ ดังนี้

$$n \xrightarrow[0]{11} c \xrightarrow[0]{01} c \xrightarrow[1]{11} c \xrightarrow[0]{10} c \xrightarrow[0]{01} c$$
$$\xrightarrow[1]{11} c \xrightarrow[0]{01} c \xrightarrow[1]{00} n \xrightarrow[1]{10} n \xrightarrow[0]{11} c \xrightarrow[1]{bb} s$$

ผลที่ได้ก็คือ 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1 จะเห็นข้อมูลที่ได้ตรงกับคำตอบ

ลองทดสอบกับผลบวกของ 10001001 + 01011010 มีคู่ลำดับจากขวาไปซ้ายที่เป็นข้อมูลเข้าคือ 10, 01, 00, 11, 01, 00, 01, 10 และ bb เริ่มต้นที่ n ได้ดังนี้

$$n \xrightarrow[1]{10} n \xrightarrow[1]{01} n \xrightarrow[0]{00} n \xrightarrow[0]{11} c \xrightarrow[0]{01} c$$

ผลที่ได้ก็คือ 1, 1, 0, 0, 0, 1, 1, 1 ตรงกับคำตอบคือ 11100011 เครื่องนี้สามารถคำนวณการบวกกันของเลขฐานสองได้ทุกนิพจน์และผลลัพธ์ถูกต้อง

๔. เมื่อออกแบบสร้างและทดสอบผลถูกต้องแล้ว จะได้ **ฟังก์ชันการผ่าน** นิยามได้ดังนี้

$$f(n,00) = n \quad , \quad f(c,00) = n$$
$$f(n,01) = n \quad , \quad f(c,01) = c$$
$$f(n,10) = n \quad , \quad f(c,10) = c$$
$$f(n,11) = c \quad , \quad f(c,11) = c$$
$$f(n,bb) = s \quad , \quad f(c,bb) = s$$

เป็นฟังก์ชัน $f : S \times \sum \to S$

๕. ได้ **ฟังก์ชันผลลัพธ์** นิยามได้ดังนี้

$$g(n,00) = 0 \ , \quad g(c,00) = 1$$
$$g(n,01) = 1 \ , \quad g(c,01) = 0$$
$$g(n,10) = 1 \ , \quad g(c,10) = 0$$
$$g(n,11) = 0 \ , \quad g(c,11) = 1$$
$$g(n,bb) = b \ , \quad g(c,bb) = 1$$

เป็นฟังก์ชัน $g : S \times \sum \to Z$

จะเห็นว่าเมื่อออกแบบและสร้างเครื่องสถานะจำกัดเสร็จ และทดสอบความถูกต้องแล้วก็สามารถใช้เครื่องนี้มาคำนวณกับทุกนิพจน์ตามจุดประสงค์ของเครื่องได้ตลอด

เอกสารอ้างอิง

๑. SEYMOUR LIPSCHUTZ, Theory and Problems of Essential Computer Mathematics, Mcgraw – Hill Book Company, 1982

๒. WOOD,D., Thoery of Computation , John Wiley & Sons,1987

๓. HOPCROFT, J. E. And J. D ULLMAN , Introduction to Automata Thoery Languages and Computation Addison – Wesley ,1990

๔. COHEN,D. I. A., Introduction to Computer Theory ,John Wiley & Sons,1991