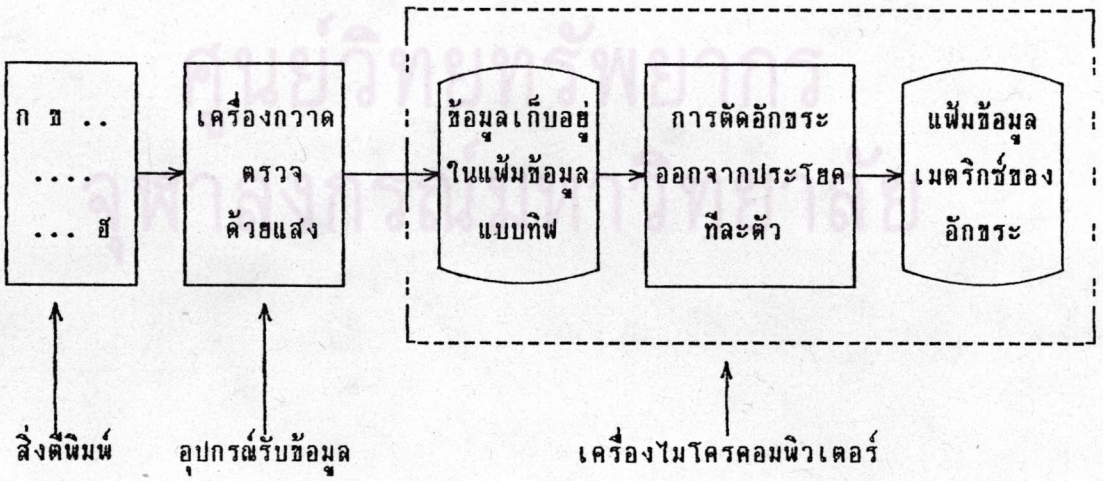
บทที่ 3

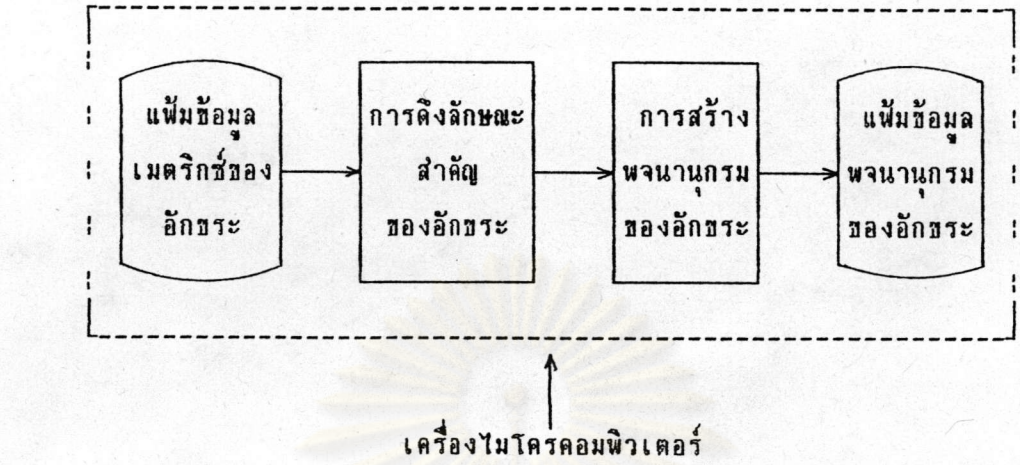
การพัฒนาโปรแกรมสำหรับการรู้จำอักขระ

การวิจัยในหัวข้อเรื่อง ระบบออฟไลน์สำหรับการรู้จำตัวพิมพ์อักขระไทยหลายรูปแบบนี้ สามารถแบ่งการทำงานออกเป็นขั้นตอนใหญ่ๆ ได้ 2 ขั้นตอน คือในขั้นตอนแรกจะเป็นการรับข้อมูลจากเครื่องกวาดตรวจด้วยแสง แล้วจึงนำข้อมูลที่ได้นั้นมาตัดอักขระออกจากประโยคทีละตัว ส่วนในขั้นตอนที่สองนั้น จะเป็นการหาลักษณะสำคัญของอักขระออกมา เพื่อนำไปใช้ในการสร้างพจนานุกรมของอักขระ รวมทั้งนำไปใช้ในการรู้จำอักขระด้วย แผนภาพแสดงการทำงานของระบบการรู้จำอักขระในการวิจัยนี้แสดงได้ดังรูปที่ 3.1(ก) และ รูปที่ 3.1(ข)

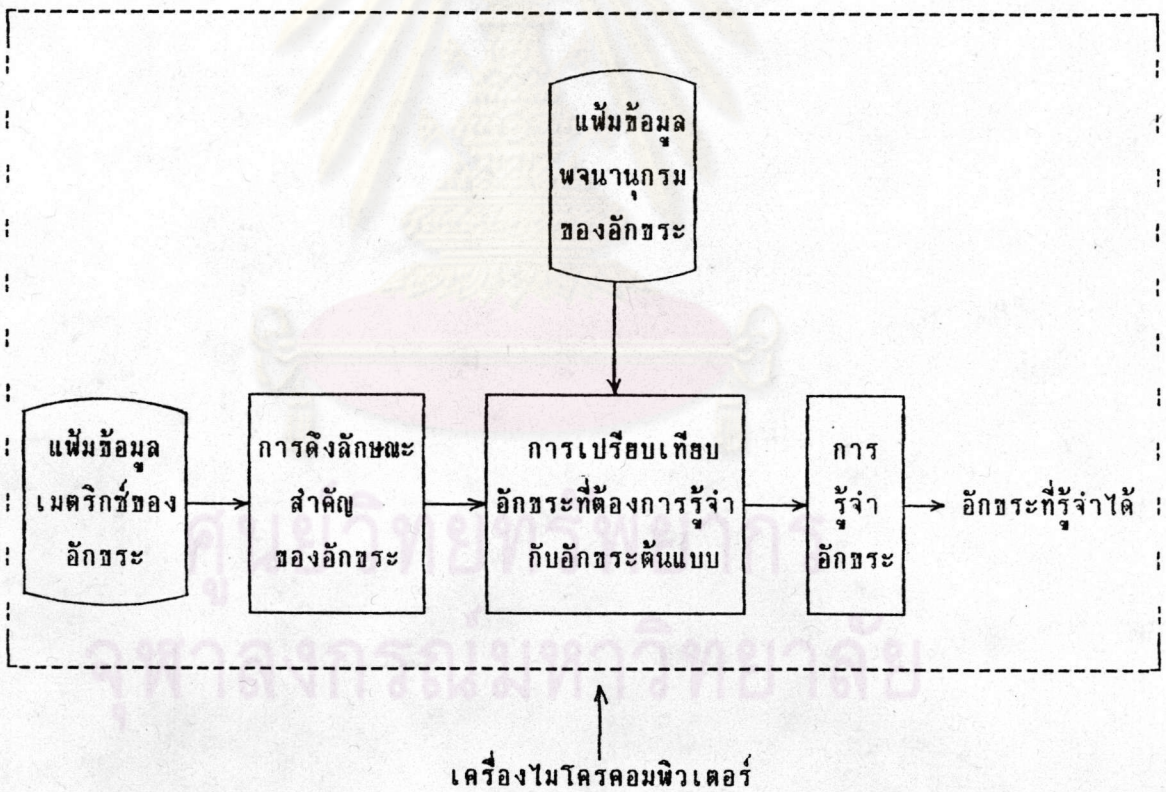
ในรูปที่ 3.1(ก) แสดงให้เห็นถึงการรับข้อมูลอักขระที่พิมพ์บนสิ่งตีพิมพ์เข้าสู่เครื่องคอมพิวเตอร์ โดยใช้เครื่องกวาดตรวจด้วยแสงเป็นอุปกรณ์รับข้อมูล แล้วจึงนำแฟ้มข้อมูลที่ได้มาตัดอักขระออกจากประโยคทีละตัว และ เก็บลงในแฟ้มข้อมูลเมตริกซ์ของอักขระอีกแฟ้มหนึ่ง ส่วนในรูปที่ 3.1(ข) เป็นการนำแฟ้มข้อมูลเมตริกซ์ของอักขระที่ได้จากขั้นตอนแรกมาประมวลผลโดยการดึงลักษณะสำคัญออกจากเมตริกซ์ของอักขระ เพื่อนำมาสร้างพจนานุกรมของอักขระ และใช้ในการรู้จำอักขระ



รูปที่ 3.1(ก) แสดงการรับข้อมูลจากเครื่องกวาดตรวจด้วยแสงและการตัดอักขระจากประโยคทีละตัว



หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.1(ข) แสดงการดึงลักษณะสำคัญของอักขระเพื่อใช้สร้างพจนานุกรมของอักขระและรู้จำอักขระ

3.1 การออกแบบ

โครงสร้างการทำงานของระบบการรู้จำอักขระในการวิจัยนี้ประกอบด้วย 5 ขั้นตอน ดังนี้

3.1.1 ขั้นตอนของการรับข้อมูลจากเครื่องกวาดตรวจด้วยแสง

3.1.2 ขั้นตอนของการตัดอักขระโดยอัตโนมัติ (segmentation stage)

3.1.3 ขั้นตอนของการหาลักษณะสำคัญของอักขระ
แบ่งเป็น 8 ขั้นตอนย่อย คือ
- 3.1.3.1 การกำจัดสัญญาณรบกวนจากเมตริกซ์ของอักขระ
- 3.1.3.2 การหาจุดเริ่มต้นของเส้นแสดงขอบของอักขระ
- 3.1.3.3 การหาเส้นแสดงขอบของอักขระพร้อมทั้งค่ารหัสทิศทาง
- 3.1.3.4 การหาจุดเปลี่ยนทิศทางบนเส้นแสดงขอบของอักขระ
- 3.1.3.5 การหาระยะห่างระหว่างจุดบ่งความนูนหรือจุดบ่งความเว้า
- 3.1.3.6 การกำหนดส่วนโค้งบนเส้นแสดงขอบของอักขระ
- 3.1.3.7 การหาจำนวนหัวของอักขระ
- 3.1.3.8 การหาอัตราส่วนความกว้างต่อความสูงของอักขระ

3.1.4 ขั้นตอนของการสร้างพจนานุกรมของอักขระ
แบ่งเป็น 2 ขั้นตอนย่อย คือ
- 3.1.4.1 การเก็บอักขระต้นแบบ
- 3.1.4.2 การเรียนรู้

3.1.5 ขั้นตอนของการรู้จำอักขระ
แบ่งเป็น 4 ขั้นตอนย่อย คือ
- 3.1.5.1 การรับอักขระที่ต้องการรู้จำเข้าสู่ระบบการรู้จำอักขระ
- 3.1.5.2 การหาลักษณะสำคัญของอักขระที่ต้องการรู้จำ
- 3.1.5.3 การเปรียบเทียบอักขระที่ต้องการรู้จำกับอักขระต้นแบบ
- 3.1.5.4 การตัดสินใจเพื่อการรู้จำอักขระ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.2 การรับข้อมูลจากเครื่องกวาดตรวจด้วยแสง

ในการวิจัยนี้ใช้เครื่องกวาดตรวจด้วยแสงชนิดที่สามารถรับข้อมูลเข้าได้ทีละ 1 หน้ากระดาษ เป็นอุปกรณ์รับข้อมูล โดยอุปกรณ์นี้จะเชื่อมต่ออยู่กับเครื่องไมโครคอมพิวเตอร์ และมีโปรแกรมชื่อ Scanning Gallery Plus 5.0 เป็นโปรแกรมสำหรับควบคุมและจัดการในการรับข้อมูลเข้ามา ข้อมูลที่ได้จากการกวาดตรวจหน้ากระดาษ 1 หน้า จะถูกเก็บไว้ในแฟ้มข้อมูลแบบบิตอิมเมจ (bit image file) โดยรูปแบบการเก็บเป็นแบบทิฟ (TIFF format) การกวาดตรวจนี้ใช้ความละเอียด 300 จุดต่อนิ้ว

ในรูปที่ 3.2 จะเป็นตัวอย่างหนึ่งของรูปแบบของอักขระที่นำมาใช้ในการวิจัย รูปแบบข้อมูลอักขระที่ใช้ในการวิจัยนี้ได้มาจากรูปแบบอักขระซึ่งมีอยู่ในโปรแกรม Pagemaker โดยขนาดของอักขระที่ใช้มีขนาดเป็น 10 และถูกพิมพ์ออกมาโดยผ่านเครื่องพิมพ์ชนิดเลเซอร์

ก ข ค ฆ ง จ ฉ ช ซ

ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด

ต ถ ท ธ น บ ป ผ ฝ

พ ฟ ภ ม ย ร ล ว ศ

ษ ส ห ฬ อ ฮ

ะ า

ิ ี ึ ื

ุ ู

เ แ โ ใ ไ ำ ั

ฤ ็ ่ ้ ๊ ๋ ์

ฯ ๆ ฦ

รูปที่ 3.2 ตัวอย่างรูปแบบของอักขระที่ใช้ในการวิจัย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.3 โปรแกรมการตัดอักขระโดยอัตโนมัติ

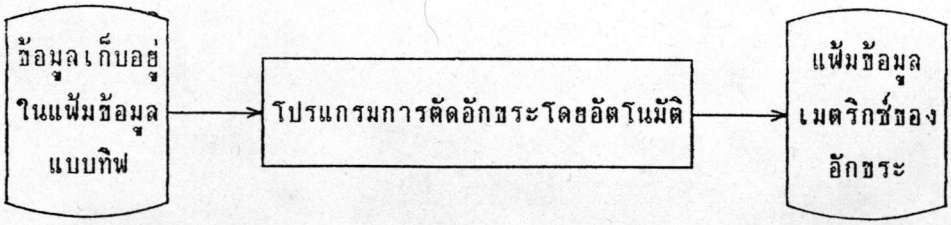
โปรแกรมนี้จะอ่านข้อมูลจากแฟ้มข้อมูลแบบบิตอิมเมจซึ่งมีรูปแบบแบบทิฟ (TIFF file) ที่ได้จากการกวาดตรวจด้วยแสงเข้ามา แล้วจึงตัดอักขระออกจากประโยคทีละตัว และเก็บอักขระแต่ละตัวที่ตัดได้ไว้ในรูปเมตริกซ์ของอักขระที่มีขนาด 40 x 40 จุด พร้อมทั้งมีการให้รหัสประจำสำหรับอักขระแต่ละตัว เพื่อใช้เป็นตัวแทนของอักขระตัวนั้นๆ ด้วย เมตริกซ์ของอักขระรวมทั้งรหัสของอักขระทั้งหมดจะถูกเก็บรวบรวมไว้ในแฟ้มข้อมูลอีกแฟ้มหนึ่ง

ในการตัดอักขระออกจากประโยคทีละตัวนี้ จะใช้ช่องว่างระหว่างอักขระเป็นเกณฑ์ในการตัดแบ่ง คือ หลังจากที่อ่านแฟ้มข้อมูลแบบทิฟ ซึ่งเก็บอักขระทั้งหมดที่พิมพ์ไว้บนสิ่งตีพิมพ์แผ่นหนึ่งเข้ามาแล้ว ในขั้นแรกก็จะตัดอักขระบรรทัดแรกที่อยู่บนหน้ากระดาษออกมาก่อน โดยใช้ช่องว่างระหว่างบรรทัดเป็นเกณฑ์ในการตัดแบ่ง จากนั้นก็จะตัดอักขระที่อยู่ในบรรทัดนั้นออกมาทีละตัว และเก็บลงในเมตริกซ์ของอักขระที่มีขนาด 40 x 40 จุด โดยในการตัดอักขระออกมานี้จะใช้ช่องว่างระหว่างอักขระแต่ละตัวเป็นเกณฑ์ในการตัดแบ่ง เมื่อตัดอักขระที่มีอยู่ในบรรทัดแรกเก็บลงในเมตริกซ์ของอักขระครบหมดทุกตัวแล้ว ก็จะใช้วิธีการเช่นเดียวกันนี้ในการตัดอักขระตัวอื่นๆ ที่อยู่ในบรรทัดที่เหลือของหน้ากระดาษออกมาจนครบหมดทุกตัว ซึ่งเมื่อตัดอักขระที่มีบนหน้ากระดาษเก็บลงเมตริกซ์ของอักขระครบหมดทุกตัวแล้ว ก็จะได้เป็นแฟ้มข้อมูลเมตริกซ์ของอักขระแฟ้มหนึ่งขึ้นมา

ในรูปที่ 3.3 จะแสดงให้เห็นถึงรูปแบบของแฟ้มข้อมูลที่เข้าสู่โปรแกรมการตัดอักขระและแฟ้มข้อมูลที่ได้หลังจากการตัดอักขระเสร็จเรียบร้อยแล้ว ซึ่งอยู่ในรูปแฟ้มข้อมูลของเมตริกซ์ของอักขระ โดยรูปแบบการเก็บของแฟ้มข้อมูลนี้เป็นดังรูปที่ 3.4

ผังงานแสดงการทำงานของโปรแกรมนี้จะแสดงได้ดังรูปที่ 3.5

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.3 รูปแบบของแฟ้มข้อมูลที่เข้าและออกจากโปรแกรมการตัดอักขระ

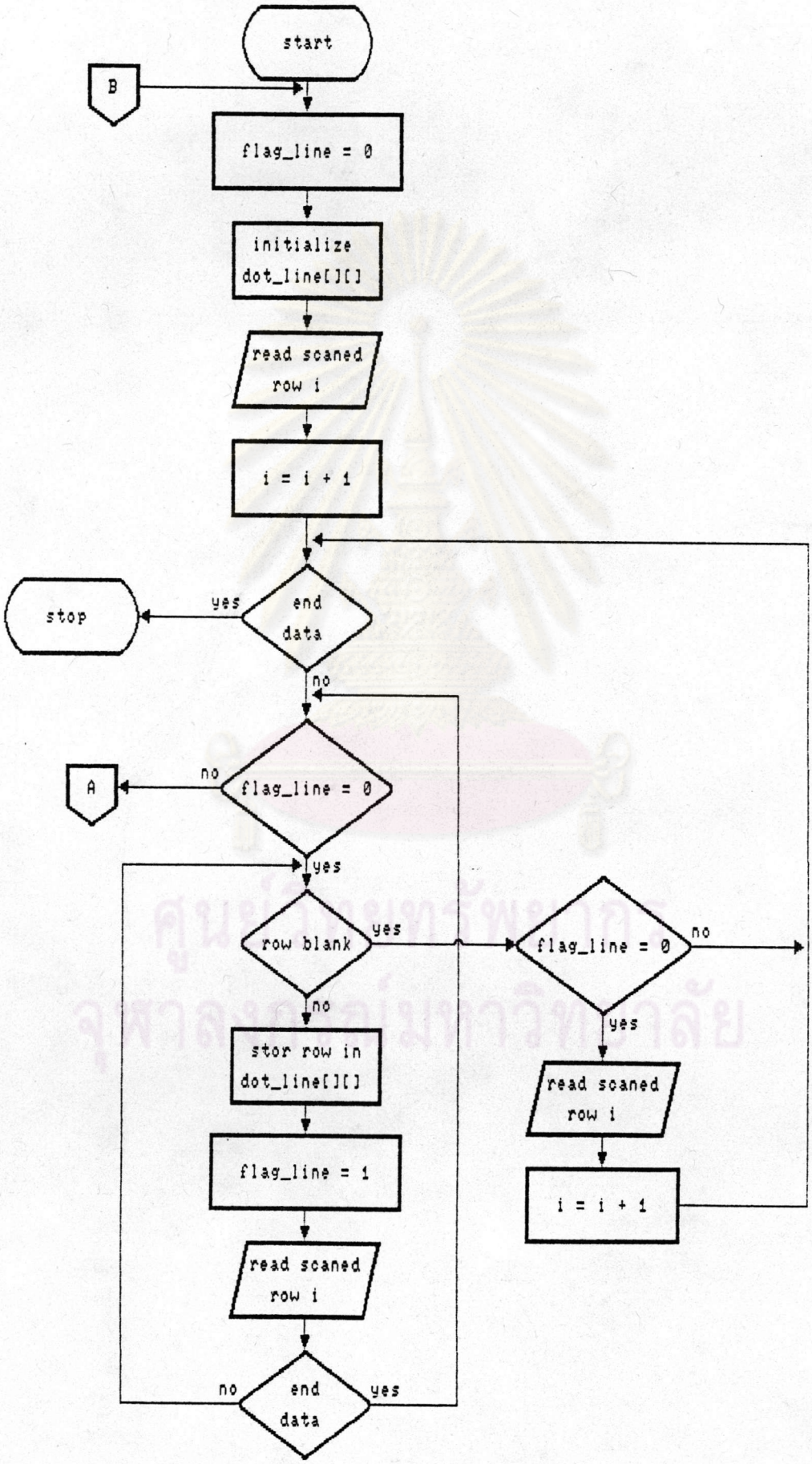
รหัสประจำอักขระ	เมตริกซ์ของอักขระ	... อักขระตัวที่ 1
รหัสประจำอักขระ	เมตริกซ์ของอักขระ	... อักขระตัวที่ 2
...	...	
...	...	
รหัสประจำอักขระ	เมตริกซ์ของอักขระ	... อักขระตัวที่ N

เมื่อ N เป็นจำนวนอักขระทั้งหมดในแฟ้มข้อมูลเมตริกซ์ของอักขระ

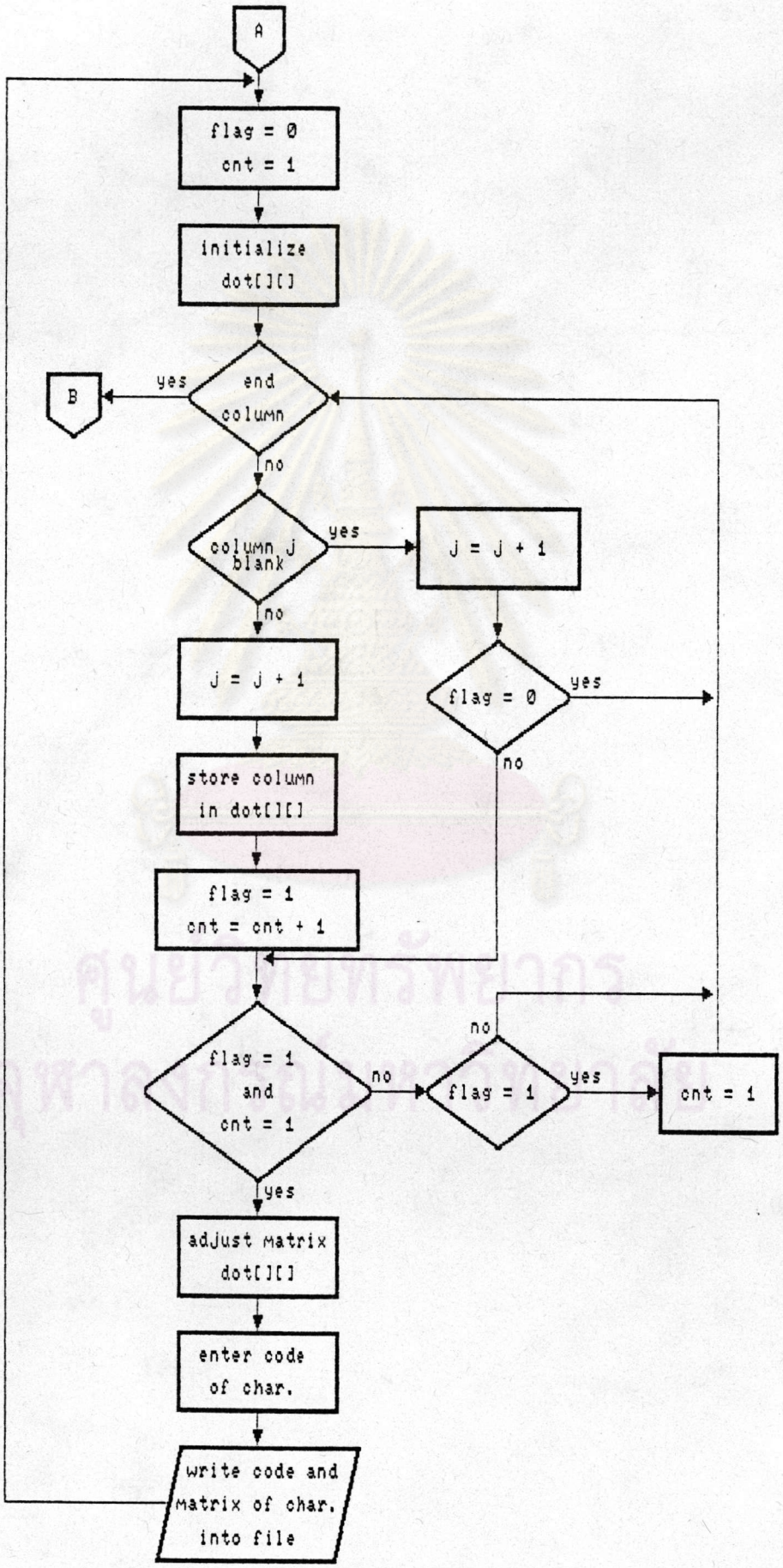
รูปที่ 3.4 รูปแบบการเก็บของแฟ้มข้อมูลเมตริกซ์ของอักขระ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 3.5 ผังงานแสดงการทำงานของโปรแกรมการตัดอักขระโดยอัตโนมัติ



รูปที่ 3.5 (ต่อ)


A
flag = 0
cnt = 1
initialize
dot[][]
end
column
yes
B
no
column j
blank
yes
j = j + 1
no
j = j + 1
flag = 0
yes
no
store column
in dot[][]
flag = 1
cnt = cnt + 1
flag = 1
and
cnt = 1
no
yes
no
flag = 1
yes
cnt = 1
adjust matrix
dot[][]
enter code
of char.
write code and
matrix of char.
into file

3.4 โปรแกรมการวิเคราะห์ลักษณะสำคัญของอักขระ

โปรแกรมนี้จะถูกเรียกใช้โดยโปรแกรมการสร้างพจนานุกรมของอักขระ และโปรแกรมการรู้จำอักขระ เนื่องจากสิ่งที่จะนำมาใช้ในการสร้างพจนานุกรมของอักขระก็คือลักษณะสำคัญของอักขระนั่นเอง นอกจากนี้แล้วก่อนที่จะสามารถรู้จำอักขระได้ก็จะต้องมีการหาลักษณะสำคัญของอักขระที่ถูกป้อนเข้ามา และ นำมาเปรียบเทียบกับลักษณะสำคัญของอักขระต้นแบบที่เก็บไว้ในพจนานุกรมของอักขระ ผังงานแสดงการทำงานของโปรแกรมนี้เป็นดังรูปที่ 3.6

ขั้นตอนการทำงานของโปรแกรมการวิเคราะห์ลักษณะสำคัญของอักขระ เป็นดังนี้

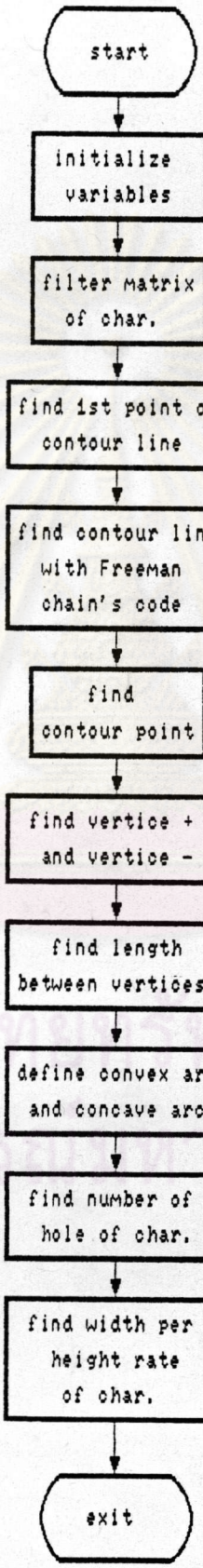
3.4.1 การกำจัดสัญญาณรบกวนจากเมตริกซ์ของอักขระ

เนื่องจากข้อมูลที่เข้ามาสู่โปรแกรมนี้จะยังเป็นข้อมูลดิบ (raw data) จึงยังคงมีสัญญาณรบกวนอยู่ที่เมตริกซ์ของอักขระดังแสดงในรูปที่ 3.7(ก) สัญญาณรบกวนนี้จะเป็นสาเหตุหนึ่งที่ทำให้เกิดความผิดพลาดในการรู้จำอักขระขึ้นได้ ดังนั้นจึงจำเป็นจะต้องกำจัดสัญญาณรบกวนออกจากเมตริกซ์ของอักขระก่อนที่จะหาลักษณะสำคัญของอักขระ เมตริกซ์ของอักขระที่กำจัดสัญญาณรบกวนออกแล้วแสดงได้ดังรูปที่ 3.7(ข)

เมตริกซ์ของอักขระเกิดจากจุดที่มีค่าเป็น 1 และจุดที่มีค่าเป็น 0 มาประกอบกันโดยจุดที่มีค่าเป็น 1 จะแทนส่วนที่มาประกอบกันเป็นรูปร่างของอักขระ และจุดที่มีค่าเป็น 0 จะแทนส่วนที่เป็นพื้นสีขาวของกระดาษ การกำจัดสัญญาณรบกวนออกจากเมตริกซ์ของอักขระทำโดยการเปรียบเทียบจุดที่มีค่าเป็น 1 ในเมตริกซ์ของอักขระกับรูปแบบที่กำหนดไว้ทีละจุด ถ้าจุดใดมีลักษณะตรงกับรูปแบบที่กำหนดจะถือว่าจุดนั้นเป็นสัญญาณรบกวน ก็จะกำจัดจุดนั้นออกไปจากเมตริกซ์ของอักขระโดยเปลี่ยนจุดที่มีค่าเป็น 1 นั้นให้มีค่าเป็น 0 ดังแสดงในรูปที่ 3.8(ก) และรูปที่ 3.8(ข) ถ้าจุดที่เป็น X มีค่าเป็น 1 ก็จะถือว่าจุดนั้นเป็นสัญญาณรบกวน ผังงานแสดงการกำจัดสัญญาณรบกวนออกจากเมตริกซ์ของอักขระเป็นดังรูปที่ 3.9

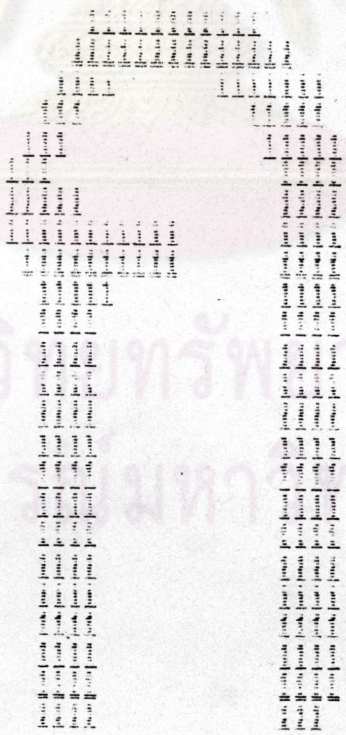
ตัวอย่างเมตริกซ์ของอักขระ ซึ่งมีสัญญาณรบกวนตรงกับรูปแบบที่กำหนดไว้ในรูปที่ 3.8(ก) และรูปที่ 3.8(ข) ได้แก่ เมตริกซ์ของอักขระในรูปที่ 3.8(ค) และรูปที่ 3.8(ง) ตามลำดับ โดยจุดที่ถือว่าเป็นสัญญาณรบกวนในเมตริกซ์ของอักขระทั้งสองนี้ ได้แก่ จุดที่มีค่าเป็น 1 ซึ่งมีเครื่องหมายวงกลมล้อมรอบอยู่

รูปที่ 3.6 ผังงานแสดงการทำงานของโปรแกรมการวิเคราะห์ลักษณะสำคัญของอักขระ



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

(ก)



(ข)

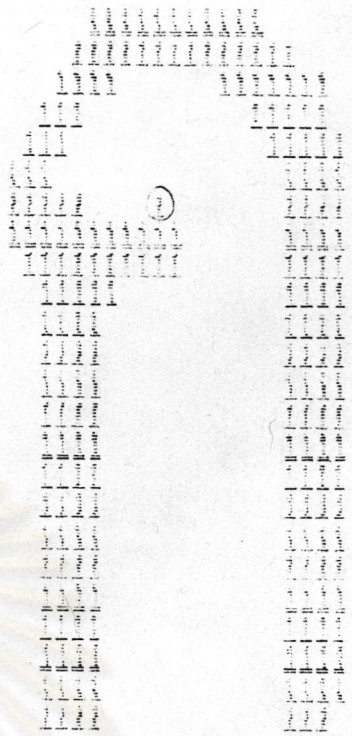
รูปที่ 3.7 (ก) เมตริกซ์ของอักขระที่ยังคงมีสัญญาณรบกวนอยู่

(ข) เมตริกซ์ของอักขระหลังจากกำจัดสัญญาณรบกวนออกแล้ว

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

0	X	0

(ก)



(ค)

	1	X	0	
		0	1	1

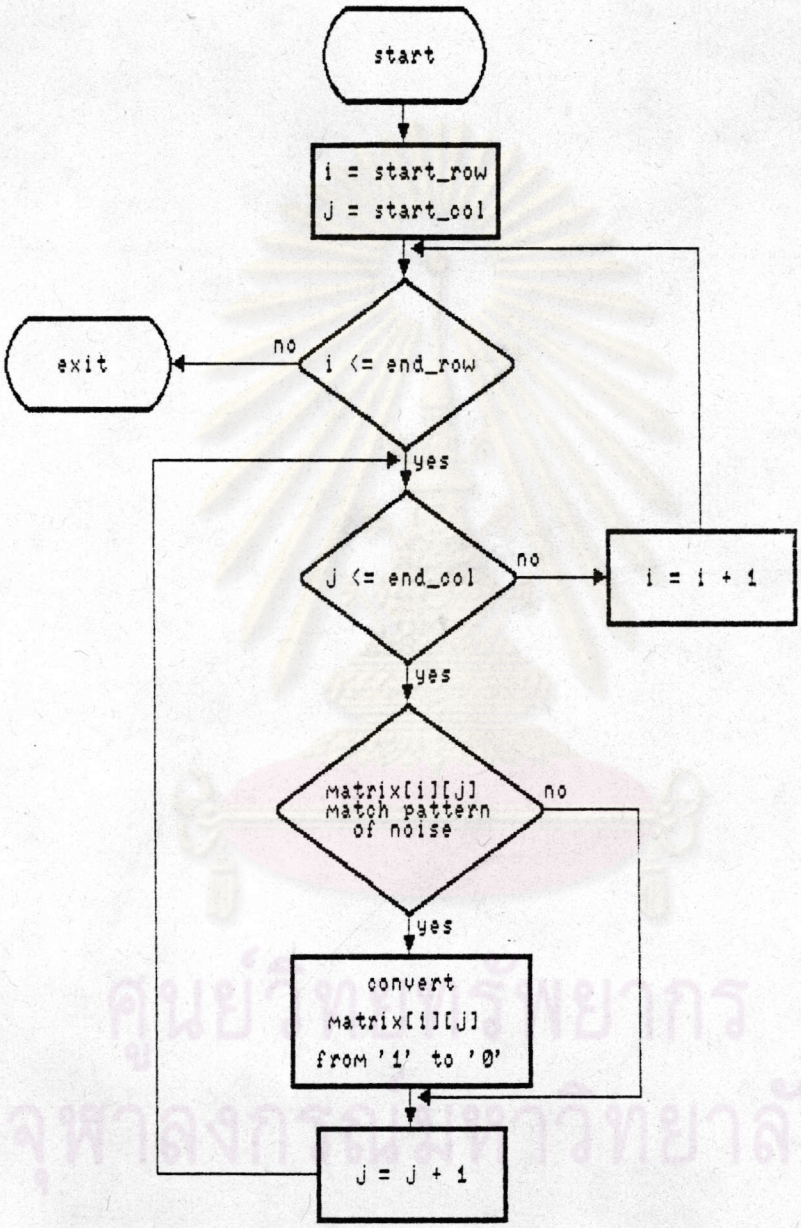
(ข)

(ง)

รูปที่ 3.8 (ก) และ (ข) รูปแบบที่ถือว่าเป็นสัญญาณรบกวน
(ค) และ (ง) เมตริกซ์ของอักขระซึ่งมีสัญญาณรบกวนตรงกับ
รูปแบบที่กำหนดในรูป (ก) และ (ข) ตามลำดับ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 3.9 ผังงานแสดงการกำจัดสัญญาณรบกวนออกจากเมตริกซ์ของอักขระ



3.4.2 การหาจุดเริ่มต้นของเส้นแสดงขอบของอักขระ

เมื่อกำจัดสัญญาณรบกวนออกจากเมตริกซ์ของอักขระแล้ว ก็จะนำเมตริกซ์ของอักขระที่ได้ใหม่นี้มาหาจุดเริ่มต้นของเส้นแสดงขอบของอักขระ โดยตรวจหาจุดที่มีค่าเป็น 1 ในเมตริกซ์ของอักขระทีละแถว (เริ่มจากแถวบนสุด) และ ภายในแถวก็จะตรวจหาทีละสดมภ์ (เริ่มจากสดมภ์ซ้ายสุด) ถ้าพบจุดที่มีค่าเป็น 1 จุดใดเป็นจุดแรก ก็จะเก็บตำแหน่ง (x,y) ของจุดนี้ไว้ในตัวแปร startx และ starty ตามลำดับ และกำหนดให้จุดนั้นเป็นจุดเริ่มต้นสำหรับใช้ในการหาจุดที่ประกอบกันเป็นเส้นแสดงขอบของอักขระจุดต่อๆ ไป

3.4.3 การหาเส้นแสดงขอบของอักขระ พร้อมทั้งค่ารหัสทิศทาง

เมื่อได้จุดเริ่มต้นของเส้นแสดงขอบของอักขระแล้ว ก็จะใช้จุดนี้สำหรับเป็นจุดเริ่มต้นในการหาจุดต่อไปที่จะมาประกอบกันเป็นเส้นแสดงขอบของอักขระ ซึ่งการหาจุดต่อไปที่จะมาประกอบกันเป็นเส้นแสดงขอบของอักขระ ทำโดยการเปรียบเทียบจุดที่อยู่รอบๆ จุดเดิมไปทีละจุดในลักษณะทวนเข็มนาฬิกา โดยตำแหน่งเริ่มต้นของจุดที่จะนำมาใช้เปรียบเทียบกับจุดเดิมคำนวณได้จากความแตกต่างของตำแหน่ง (x,y) ระหว่าง จุดที่ได้ก่อนหน้าที่จะถึงจุดเดิม 1 จุด กับจุดเดิม ดังนี้

กำหนดให้ $p(i) = (x_i, y_i)$ เป็นจุดที่ i

และ $p(i-1) = (x_{i-1}, y_{i-1})$ เป็นจุดที่ i-1

ดังนั้นความแตกต่างของตำแหน่ง (x,y) ระหว่างจุด p(i-1) กับจุด p(i) คำนวณได้ดังนี้

$$dx = x_{i-1} - x_i$$

$$dy = y_{i-1} - y_i$$

และ ถ้า dx = 0 และ dy = 1 ก็จะให้ num = 1
ถ้า dx = -1 และ dy = 1 ก็จะให้ num = 2
ถ้า dx = -1 และ dy = 0 ก็จะให้ num = 3
ถ้า dx = -1 และ dy = -1 ก็จะให้ num = 4
ถ้า dx = 0 และ dy = -1 ก็จะให้ num = 5
ถ้า dx = 1 และ dy = -1 ก็จะให้ num = 6
ถ้า dx = 1 และ dy = 0 ก็จะให้ num = 7
ถ้า dx = 1 และ dy = 1 ก็จะให้ num = 8

ดังนั้นเมื่อพิจารณาข้อกำหนดที่กล่าวมาข้างต้น ประกอบกันกับรูปที่ 3.10 ก็จะเห็นได้ว่า ถ้าจุด p(i-1) อยู่ในตำแหน่งที่มีค่า dx = -1 และ dy = 0 แล้ว ตำแหน่งเริ่มต้นของจุดที่จะนำมาใช้ในการเปรียบเทียบกับจุด p(i) เพื่อหาจุดที่จะมาประกอบกันเป็นเส้นแสดงขอบของอักขระจุดต่อไป ก็จะได้แก่ ตำแหน่งที่มีค่า num = 3 หรือถ้าจุด p(i-1) อยู่ในตำแหน่งที่มีค่า dx = 0 และ dy = -1 แล้ว ตำแหน่งเริ่มต้นของจุดที่จะนำมาใช้ในการเปรียบเทียบกับจุด p(i) ก็จะได้แก่ ตำแหน่งที่มีค่า num = 5

สำหรับในการหาตำแหน่งเริ่มต้นของจุดที่จะนำมาใช้ในการเปรียบเทียบกับจุดเริ่มต้นของเส้นแสดงขอบของอักขระ (จุด p(i) เมื่อ i = 1) นั้น จะกำหนดให้ dx = -1 และ dy = -1 ดังนั้นตำแหน่งเริ่มต้นของจุดที่จะนำมาใช้ในการเปรียบเทียบกับจุด p(1) ก็จะได้แก่ ตำแหน่งที่มีค่า num = 4

dx = -1 dy = -1	dx = -1 dy = 0	dx = -1 dy = 1
dx = 0 dy = -1	p(i) = (x_i, y_i)	dx = 0 dy = 1
dx = 1 dy = -1	dx = 1 dy = 0	dx = 1 dy = 1

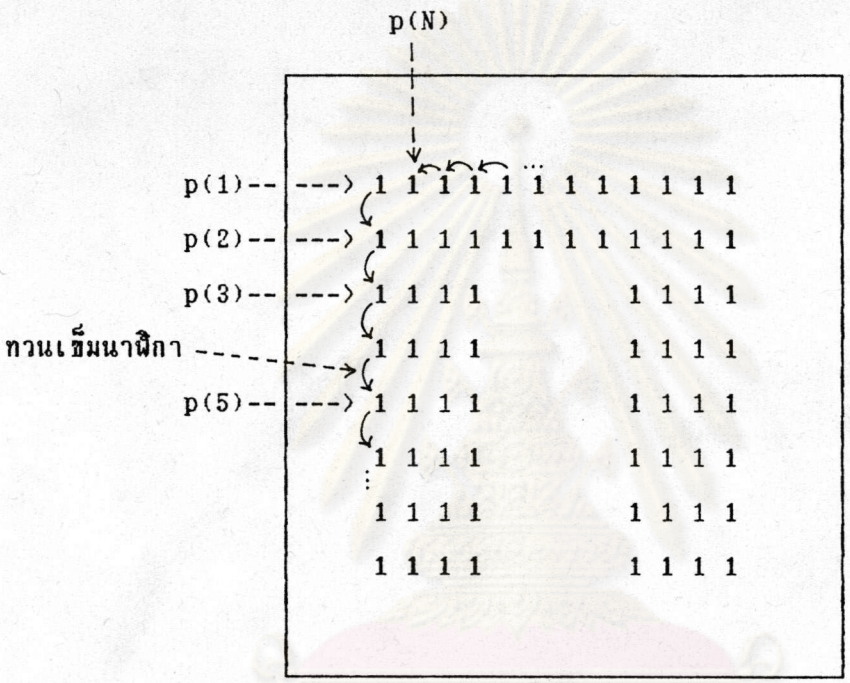
num = 3	num = 2	num = 1
num = 4	p(i)	num = 8
num = 5	num = 6	num = 7

รูปที่ 3.10 การหาตำแหน่งเริ่มต้นสำหรับใช้ในการเปรียบเทียบเพื่อหาจุดที่จะมาประกอบกันเป็นเส้นแสดงขอบของอักขระจุดต่อไป

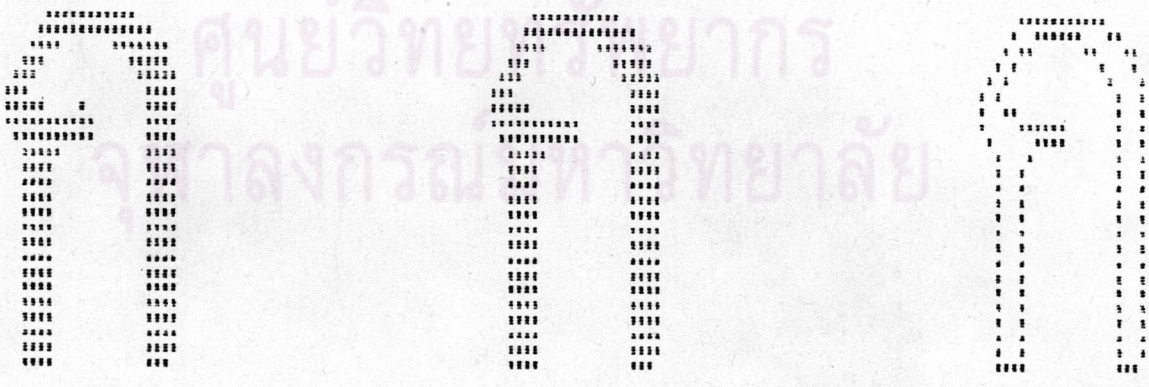
เมื่อทราบตำแหน่งเริ่มต้นของจุดที่จะนำมาใช้ในการเปรียบเทียบกับจุด p(1) แล้ว ก็จะใช้จุดนี้เป็นจุดเริ่มต้นในการเปรียบเทียบเพื่อหาจุดที่จะมาประกอบกันเป็นเส้นแสดงขอบของอักขระจุดต่อไป โดยจะเปรียบเทียบจุดที่อยู่รอบๆ จุด p(1) ไปทีละจุดในลักษณะทวนเข็มนาฬิกา จนกว่าจะพบจุดที่เป็น 1 จุดแรกเมื่อใดก็แสดงว่า จุดนั้นเป็นจุดที่จะมาประกอบกันเป็นเส้นแสดงขอบของอักขระจุดต่อไป (จุด p(2)) และเมื่อได้จุด p(2) แล้ว ก็สามารถจะหาจุดถัดไปได้ในลักษณะเดียวกันจนครบรอบ คือ วนมาพบจุดที่เป็นจุดเริ่มต้นของเส้นแสดงขอบของอักขระ (จุด p(1)) อีกครั้งหนึ่ง จึงจะหยุดการหา นั่นคือ จะได้เส้นแสดงขอบของอักขระที่สมบูรณ์

ในรูปที่ 3.11 จะแสดงให้เห็น จุดที่ประกอบกันเป็นเส้นแสดงขอบของอักขระ ซึ่งได้แก่ จุดที่มีค่าเป็น 1 (จุดเข้ม) ที่อยู่เรียงต่อกัน

และในรูปที่ 3.12 จะแสดงผลของการกำจัดสัญญาณรบกวนของอักขระที่ได้จากการกวาดตรวจด้วยเครื่องกวาดตรวจด้วยแสง และผลของการหาเส้นแสดงขอบของอักขระ



รูปที่ 3.11 การหาจุดถัดไปแต่ละจุดในทิศทวนเข็มนาฬิกาเพื่อหาเส้นแสดงขอบของอักขระ



รูปที่ 3.12 (ก) อักขระที่ได้จากการกวาดตรวจด้วยเครื่องกวาดตรวจด้วยแสง
(ข) อักขระที่ผ่านการกำจัดสัญญาณรบกวนออกจากเมตริกซ์ของอักขระแล้ว
(ค) เส้นแสดงขอบของอักขระ

จากที่กล่าวมาทั้งหมดข้างต้นนั้น สามารถเขียนเป็นสัญญลักษณ์ที่ใช้แสดงถึงเส้นแสดงขอบของอักขระในลักษณะสมการได้ดังนี้

กำหนด p(n) เป็น จุดที่อยู่บนเส้นแสดงขอบของอักขระ

โดย $p(n) = (p_{xn}, p_{yn})$ เป็นจุดที่ n บนเส้นแสดงขอบของอักขระ

เมื่อ $n = 1, \ldots, N$

และ N เป็นจำนวนจุดทั้งหมดบนเส้นแสดงขอบของอักขระ

จากข้อกำหนดของตัวแปรที่กล่าวมา จะได้ว่า

$p(1) = (p_{x1}, p_{y1})$ เป็นจุดที่ 1 หรือจุดเริ่มต้นบนเส้นแสดงขอบของอักขระ

โดย p_{x1} = startx

p_{y1} = starty

ในขณะที่หาจุดที่ประกอบกันเป็นเส้นแสดงขอบของอักขระไปทีละจุดนั้น ก็จะได้รหัสทิศทางของแต่ละจุดไปพร้อมกันด้วย รหัสทิศทางของแต่ละจุดนี้สามารถจะหาได้โดยการเปรียบเทียบค่าความแตกต่างของตำแหน่ง (x,y) ระหว่างจุดที่ได้ใหม่กับจุดเดิม ดังนี้

กำหนด $p(i) = (x_i, y_i)$ เป็นจุดที่ i (จุดเดิม)

และ $p(i+1) = (x_{i+1}, y_{i+1})$ เป็นจุดที่ i + 1 (จุดที่ได้ใหม่)

จะได้ว่า $incx = x_{i+1} - x_i$

$incy = y_{i+1} - y_i$

ดังนั้น ถ้า incx = 0 และ incy = 1 ก็ให้ chain_code = 0

ถ้า incx = -1 และ incy = 1 ก็ให้ chain_code = 1

ถ้า incx = -1 และ incy = 0 ก็ให้ chain_code = 2

ถ้า incx = -1 และ incy = -1 ก็ให้ chain_code = 3

ถ้า incx = 0 และ incy = -1 ก็ให้ chain_code = 4

ถ้า incx = 1 และ incy = -1 ก็ให้ chain_code = 5

ถ้า incx = 1 และ incy = 0 ก็ให้ chain_code = 6

ถ้า incx = 1 และ incy = 1 ก็ให้ chain_code = 7

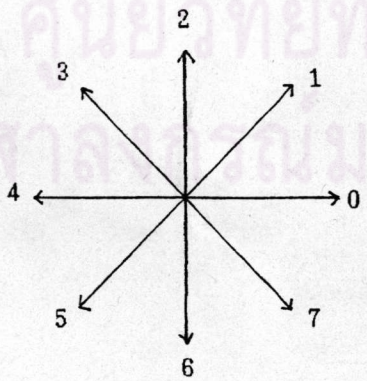
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ดังนั้นจะเห็นได้ว่า การกำหนดรหัสทิศทาง (chain_code) ให้กับแต่ละจุดบนเส้นแสดงขอบของอักขระนี้จะขึ้นกับทิศทางของจุดที่เปลี่ยนไป ซึ่งสามารถแสดงได้ดังรูปที่ 3.13 ในรูปที่ 3.13 นี้ จะเห็นได้ว่า ถ้าจุด p(i+1) อยู่ในตำแหน่งที่มีค่า incx = -1 และ incy = -1 แล้ว ค่ารหัสทิศทางที่มีค่าเป็น 3 ก็จะถูกกำหนดให้กับจุด p(i) หรือถ้า incx = 1 และ incy = 1 แล้ว ค่ารหัสทิศทางที่มีค่าเป็น 7 ก็จะถูกกำหนดให้กับจุด p(i) ตัวอย่างเส้นแสดงขอบของอักขระ พร้อมทั้งค่ารหัสทิศทางซึ่งกำกับอยู่ที่แต่ละจุดที่ประกอบกันเป็นเส้นแสดงขอบของอักขระ (จุดเข้ม) แสดงได้ดังรูปที่ 3.14

ผังงานแสดงการหาเส้นแสดงขอบของอักขระพร้อมทั้งค่ารหัสทิศทาง แสดงได้ดังรูปที่ 3.15

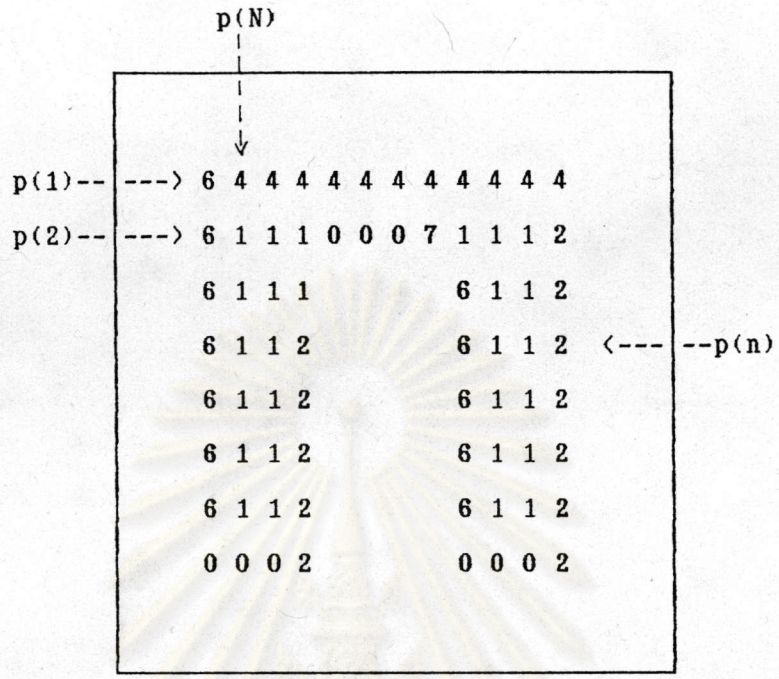
incx = -1 incy = -1	incx = -1 incy = 0	incx = -1 incy = 1
incx = 0 incy = -1	p(i) = (x_i, y_i)	incx = 0 incy = 1
incx = 1 incy = -1	incx = 1 incy = 0	incx = 1 incy = 1

รหัส = 3	รหัส = 2	รหัส = 1
รหัส = 4	p(i)	รหัส = 0
รหัส = 5	รหัส = 6	รหัส = 7



รหัสแบบลูกโซ่ของฟรีแมน

รูปที่ 3.13 การกำหนดค่ารหัสทิศทางให้กับจุดบนเส้นแสดงขอบของอักขระ

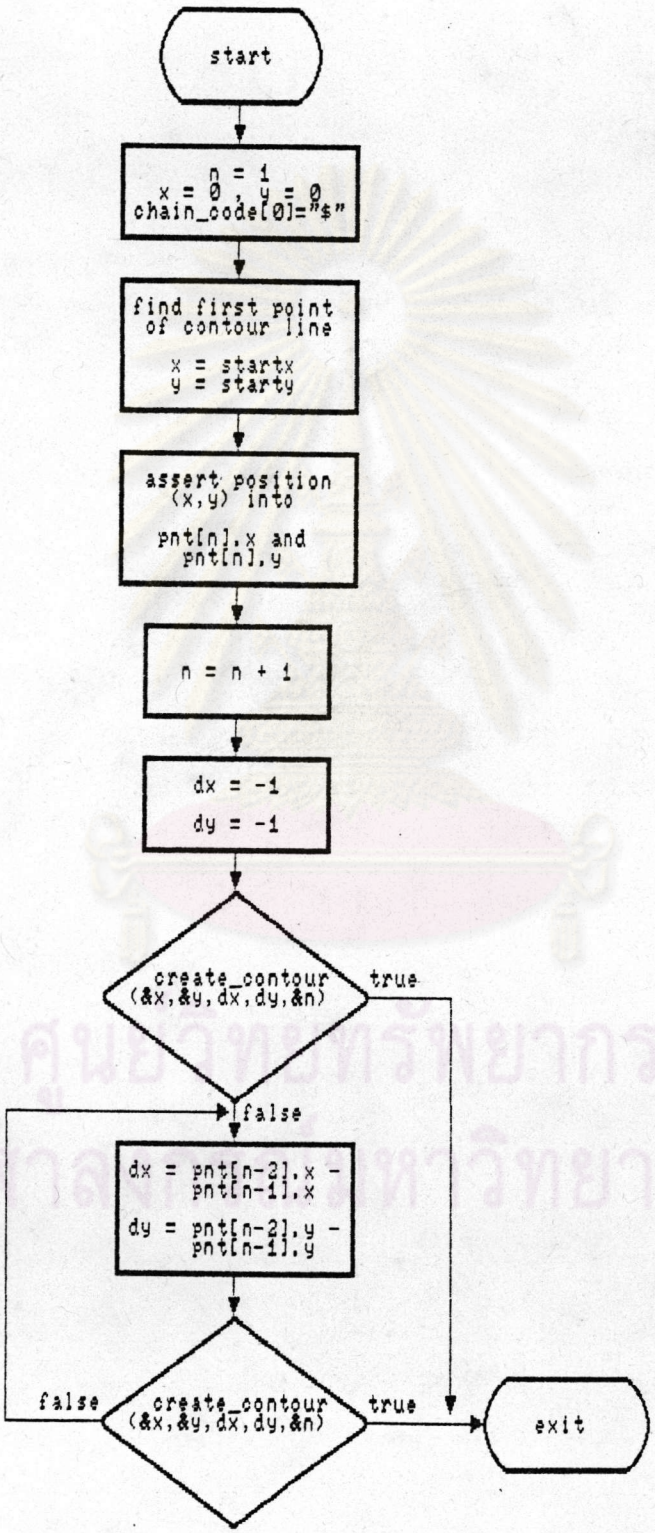


รูปที่ 3.14 ตัวอย่างของเส้นแสดงขอบของอักขระ (จุดเข้ม) พร้อมทั้งค่ารหัสทิศทาง

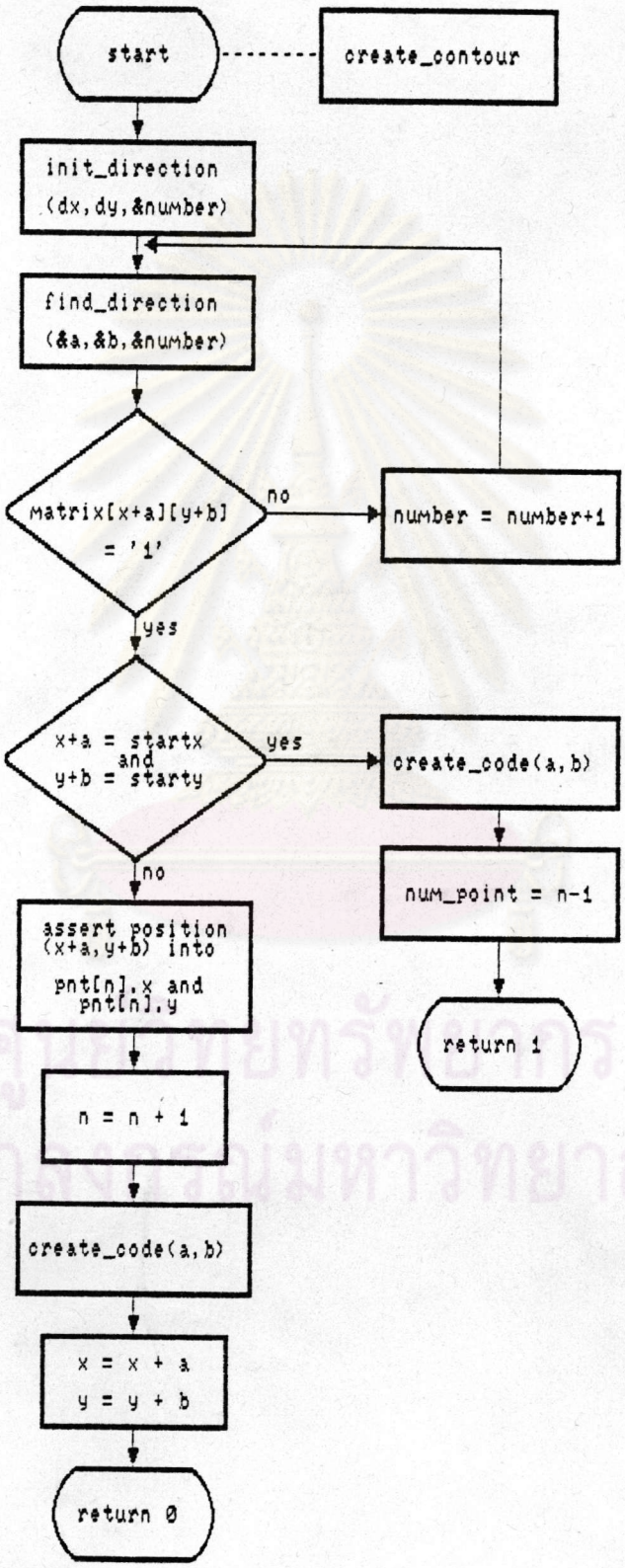
จากรูปที่ 3.14 จะได้ว่า $p(n) = (p_{xn}, p_{yn})$ (เมื่อ $n = 1, \ldots, N$ และ N เป็นจำนวนจุดทั้งหมดบนเส้นแสดงขอบของอักขระ) เป็นจุดซึ่งมีค่ารหัสแบบลูกโซ่ของฟรีแมน $F_4 = \{ 0,1,2,3,4,5,6,7 \}$ กำกับอยู่

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

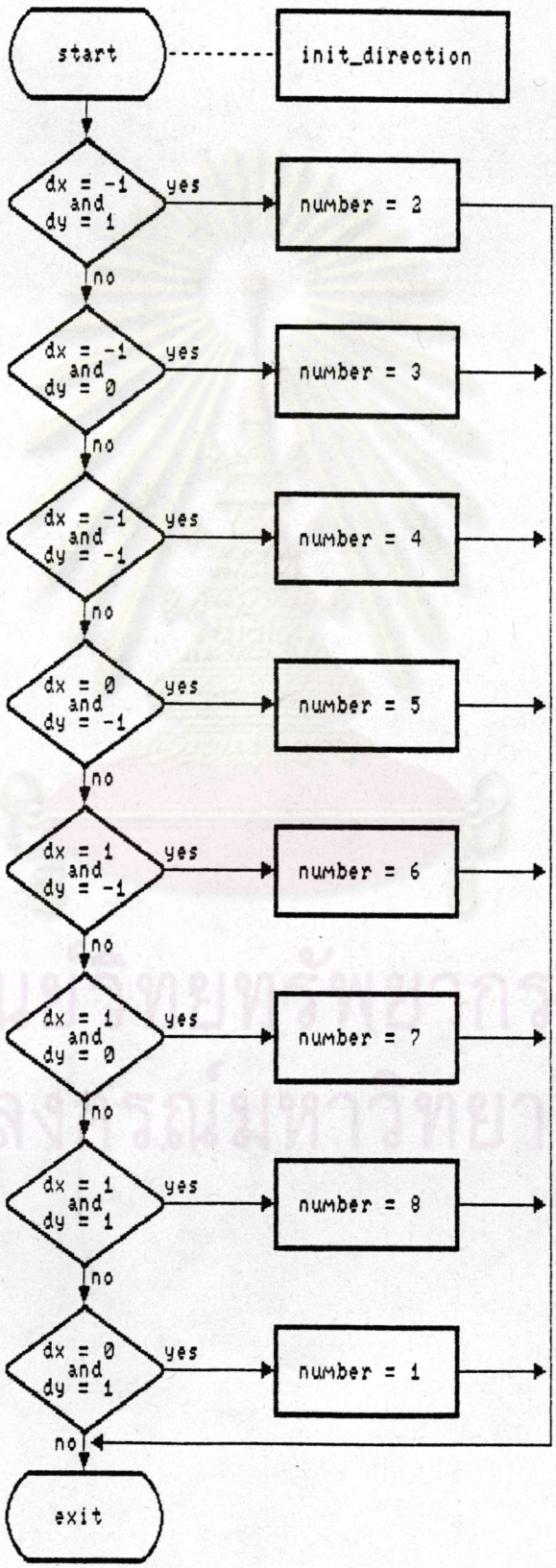
รูปที่ 3.15 ผังงานแสดงการหาเส้นแสดงขอบของอักขระพร้อมทั้งค่ารหัสทิศทาง



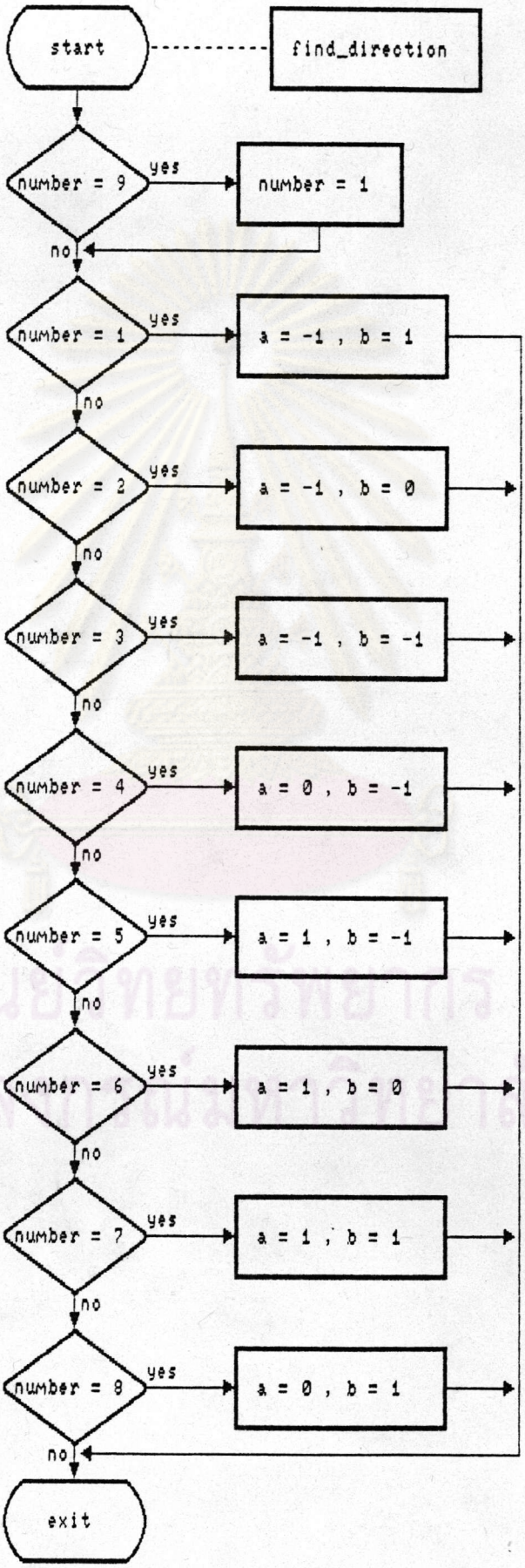
รูปที่ 3.15 (ต่อ)


start
create_contour
init_direction
(dx,dy,&number)
find_direction
(&a,&b,&number)
matrix[x+a][y+b]
= '1'
no
number = number+1
yes
x+a = startx
and
y+b = starty
yes
create_code(a,b)
num_point = n-1
return 1
no
assert position
(x+a,y+b) into
pnt[n].x and
pnt[n].y
n = n + 1
create_code(a,b)
x = x + a
y = y + b
return 0

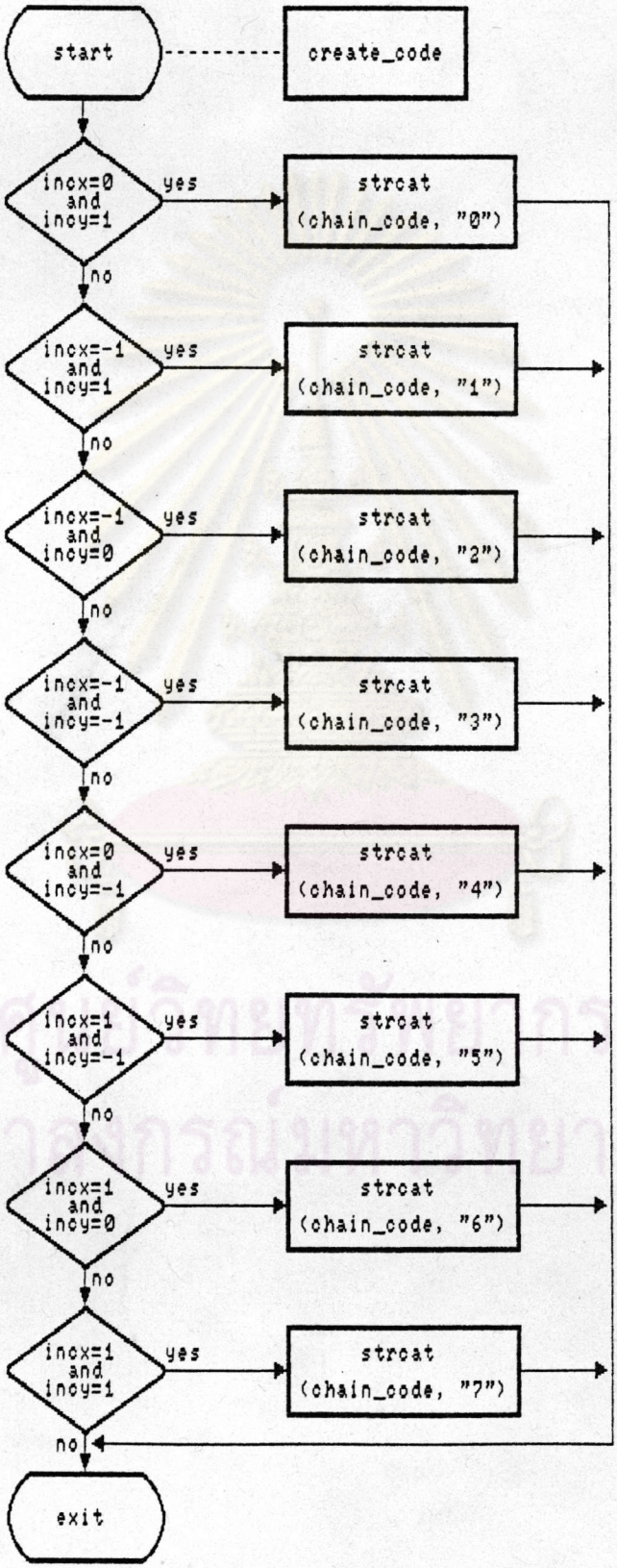
รูปที่ 3.15 (ต่อ)


start
init_direction
dx = -1 and dy = 1
yes
number = 2
no
dx = -1 and dy = 0
yes
number = 3
no
dx = -1 and dy = -1
yes
number = 4
no
dx = 0 and dy = -1
yes
number = 5
no
dx = 1 and dy = -1
yes
number = 6
no
dx = 1 and dy = 0
yes
number = 7
no
dx = 1 and dy = 1
yes
number = 8
no
dx = 0 and dy = 1
yes
number = 1
no
exit

รูปที่ 3.15 (ต่อ)

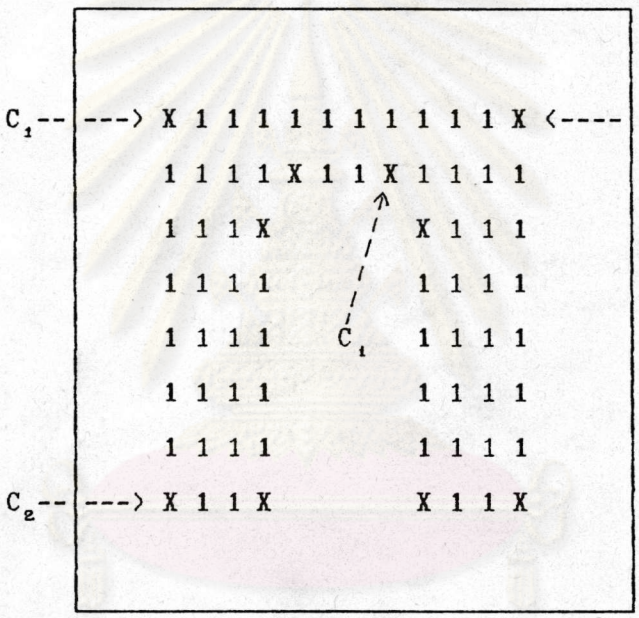

start
find_direction
number = 9
yes
number = 1
no
number = 1
yes
a = -1 , b = 1
no
number = 2
yes
a = -1 , b = 0
no
number = 3
yes
a = -1 , b = -1
no
number = 4
yes
a = 0 , b = -1
no
number = 5
yes
a = 1 , b = -1
no
number = 6
yes
a = 1 , b = 0
no
number = 7
yes
a = 1 , b = 1
no
number = 8
yes
a = 0 , b = 1
no
exit

รูปที่ 3.15 (ต่อ)


start
create_code
incx=0 and incy=1
yes
strcat (chain_code, "0")
no
incx=-1 and incy=1
yes
strcat (chain_code, "1")
no
incx=-1 and incy=0
yes
strcat (chain_code, "2")
no
incx=-1 and incy=-1
yes
strcat (chain_code, "3")
no
incx=0 and incy=-1
yes
strcat (chain_code, "4")
no
incx=1 and incy=-1
yes
strcat (chain_code, "5")
no
incx=1 and incy=0
yes
strcat (chain_code, "6")
no
incx=1 and incy=1
yes
strcat (chain_code, "7")
no
exit

3.4.4 การหาจุดเปลี่ยนทิศทางบนเส้นแสดงขอบของอักขระ

เมื่อได้เส้นแสดงขอบของอักขระ พร้อมทั้งค่ารหัสทิศทางกำกับที่แต่ละจุดที่ประกอบกันเป็นเส้นแสดงขอบของอักขระแล้ว ต่อไปก็จะเป็นการหาจุดที่มีการเปลี่ยนทิศทางบนเส้นแสดงขอบของอักขระ ซึ่งสามารถจะหาได้โดยการเปรียบเทียบค่ารหัสทิศทางซึ่งกำกับอยู่ที่แต่ละจุดบนเส้นแสดงขอบของอักขระไปทีละจุด ถ้าพบว่าจุดใดมีค่ารหัสทิศทางที่เปลี่ยนไปก็จะกำหนดให้จุดนั้นเป็นจุดเปลี่ยนทิศทาง C_i ในรูปที่ 3.16 จะเห็นว่าจุดเปลี่ยนทิศทางบนเส้นแสดงขอบของอักขระ ได้แก่ จุดที่แสดงด้วยตัวอักษร X



รูปที่ 3.16 จุดเปลี่ยนทิศทางบนเส้นแสดงขอบของอักขระ (จุด X)

กำหนดให้ $C_i = (C_{xi}, C_{yi})$: $i = 1, \ldots, I$

โดย C_i เป็นลำดับของจุดที่มีการเปลี่ยนทิศทางตามรหัสแบบลูกโซ่ของฟรีแมน

และ I เป็นจำนวนจุดเปลี่ยนทิศทางทั้งหมดบนเส้นแสดงขอบของอักขระ

กำหนด $F_i \in \{0, 1, 2, 3, 4, 5, 6, 7\}$

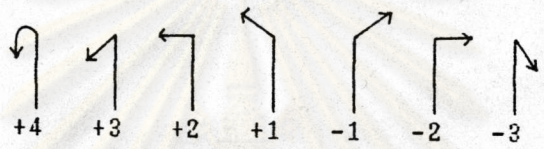
โดย F_i เป็นค่ารหัสแบบลูกโซ่ของฟรีแมนของจุด C_i

และ $S_i \in \{1, 2, 3, 4, -1, -2, -3\}$

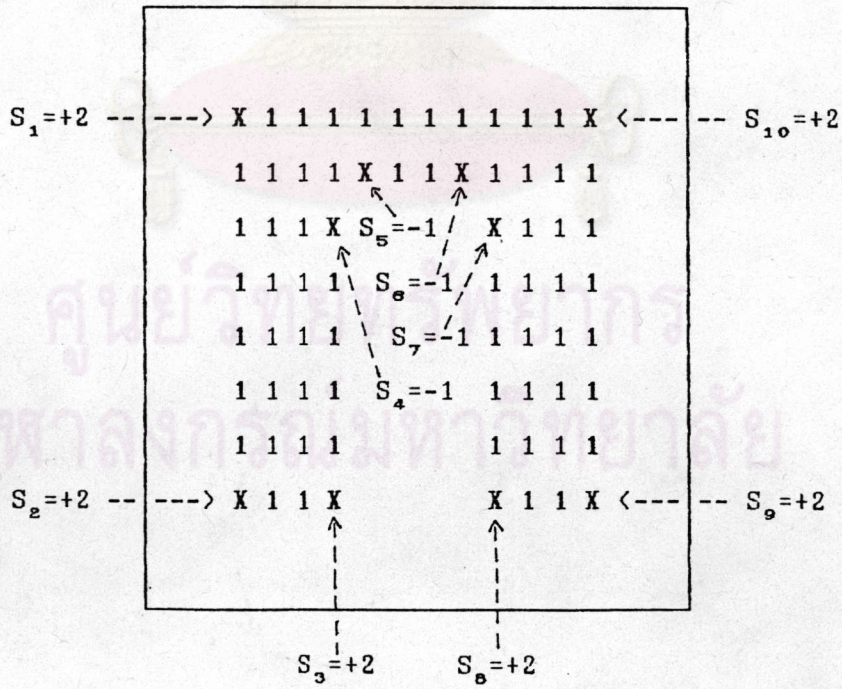
โดย S_i เป็นเครื่องหมายซึ่งจะกำหนดให้กับจุด C_i แต่ละจุด ตามลักษณะการเปลี่ยนทิศทาง ดังแสดงในรูปที่ 3.17

จากนั้นก็จะเป็นการกำหนดเครื่องหมาย S_i ให้กับจุดเปลี่ยนทิศทางแต่ละจุดตามลักษณะการเปลี่ยนทิศทางว่าตามเข็มหรือทวนเข็มนาฬิกาดังในรูปที่ 3.17 ก็จะได้เมตริกซ์ของอักขระที่มีเครื่องหมาย S_i กำกับอยู่ที่จุดเปลี่ยนทิศทางดังในรูปที่ 3.18

การกำหนดเครื่องหมาย S_i ให้กับจุดเปลี่ยนทิศทาง C_i แต่ละจุดนั้นจะพิจารณาจากความแตกต่างระหว่างค่ารหัสทิศทาง F_i ของจุดเปลี่ยนทิศทาง 2 จุดที่อยู่ติดกัน คือ จุด C_i และ จุด C_{i-1} ดังผังงานแสดงการทำงานในรูปที่ 3.19

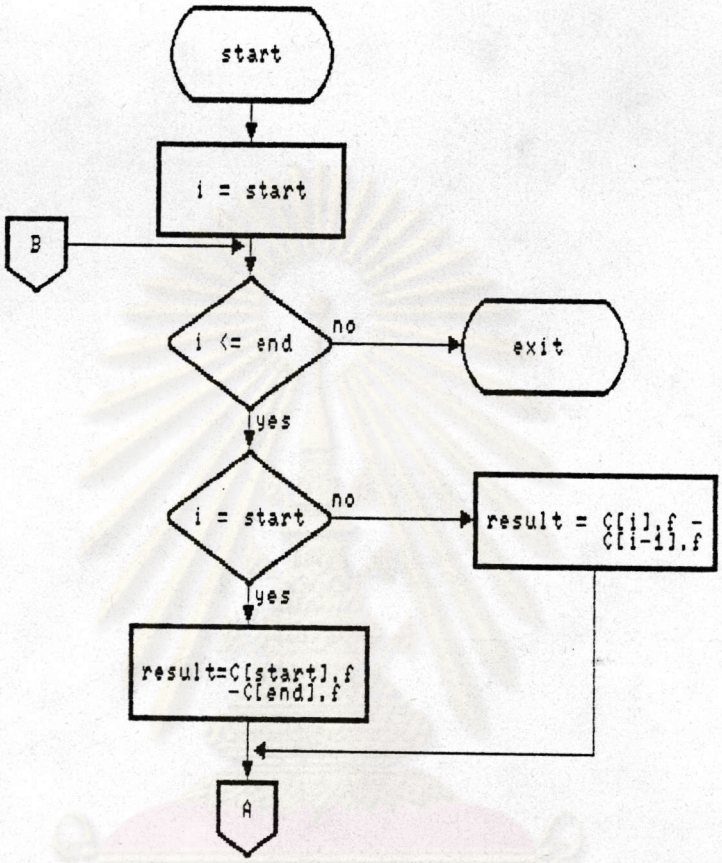


รูปที่ 3.17 เครื่องหมาย S_i



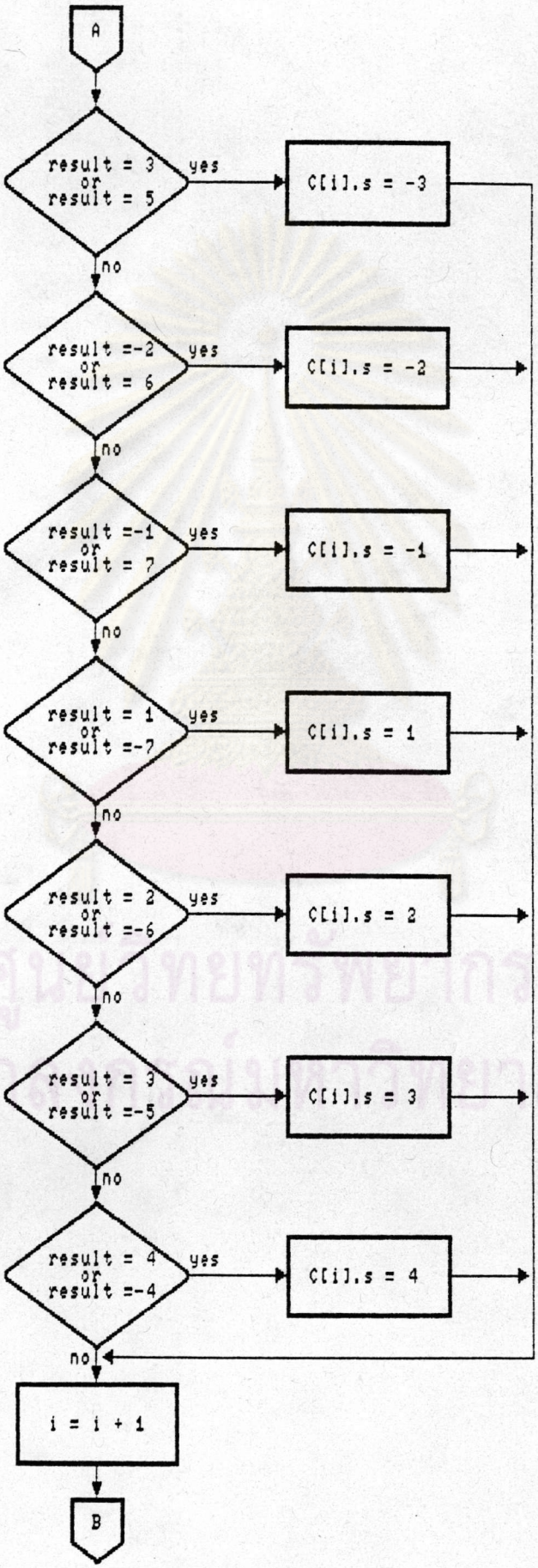
รูปที่ 3.18 จุดเปลี่ยนทิศทาง (จุด X) ซึ่งมีเครื่องหมาย S_i กำกับอยู่

รูปที่ 3.19 ผังงานแสดงการกำหนดเครื่องหมาย S_i ให้กับจุดเปลี่ยนทิศทาง C_i แต่ละจุด



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 3.19 (ต่อ)


A
result = 3 or result = 5
yes
C[i].s = -3
no
result =-2 or result = 6
yes
C[i].s = -2
no
result =-1 or result = 7
yes
C[i].s = -1
no
result = 1 or result =-7
yes
C[i].s = 1
no
result = 2 or result =-6
yes
C[i].s = 2
no
result = 3 or result =-5
yes
C[i].s = 3
no
result = 4 or result =-4
yes
C[i].s = 4
no
i = i + 1
B

จากนั้นก็จะนำเครื่องหมาย S_i พร้อมด้วยค่ารหัสทิศทาง F_i ซึ่งกำกับอยู่ที่จุดเปลี่ยนทิศทาง C_i แต่ละจุด มาใช้ในการกำหนดจุดบ่งความนูนและจุดบ่งความเว้าบนเส้นแสดงขอบของอักขระ ดังผังงานแสดงการทำงานในรูปที่ 3.20 การกำหนดจุดบ่งความนูนและจุดบ่งความเว้าให้กับเส้นแสดงขอบของอักขระนี้ จะกำหนดโดยใช้ข้อกำหนด 2 ข้อ ดังนี้

ข้อกำหนดที่ 1 : ถ้าจุด C_i สอดคล้องกับเงื่อนไขใดเงื่อนไขหนึ่ง ดังต่อไปนี้ แล้ว จุด C_i จุดนั้น ก็จะถูกกำหนดให้เป็น จุดบ่งความนูน (vertice +)

1.1 $(S_i > 0)$ และ $(S_{i-1} > 0$ หรือ $S_{i+1} > 0)$

1.2 $(S_i > 0)$ และ $(S_{i-1} < 0)$ และ $(S_{i+1} < 0)$ และ $(F_i \neq FF)$ และ $(DD_{i,i+1} \geq K1)$

โดย $DD_{i,i+1}$ เป็นระยะห่างระหว่างจุด C_i และ จุด C_{i+1}

เมื่อ $DD_{i,i+1}$ = (จำนวนจุดซึ่งอยู่ระหว่างจุดที่ i และ จุดที่ i+1) + 1

และ FF เป็นค่ารหัสแบบลูกโซ่ของฟรีแมนของจุดซึ่งตรวจพบก่อนที่จะถึงจุด C_i

K1 เป็นค่าคงที่ ซึ่งได้จากการทดลองกับข้อมูลทดสอบ

ถ้าจุด C_i ถูกกำหนดให้เป็นจุดบ่งความนูน และ $DD_{i,i+1}$ มีค่ามากกว่า K2 แล้ว ค่า FF จะถูกกำหนดให้มีค่าใหม่ตามค่าของ F_i โดย FF จะถูกกำหนดให้มีค่าเริ่มต้นเป็น 4 และ K2 เป็นค่าคงที่ ซึ่งได้จากการทดลองกับข้อมูลทดสอบ

ข้อกำหนดที่ 2 : ถ้าจุด C_i สอดคล้องกับเงื่อนไขใดเงื่อนไขหนึ่ง ดังต่อไปนี้ แล้ว จุด C_i จุดนั้น ก็จะถูกกำหนดให้เป็น จุดบ่งความเว้า (vertice -)

2.1 $(S_i < 0)$ และ $(S_{i-1} < 0$ หรือ $S_{i+1} < 0)$

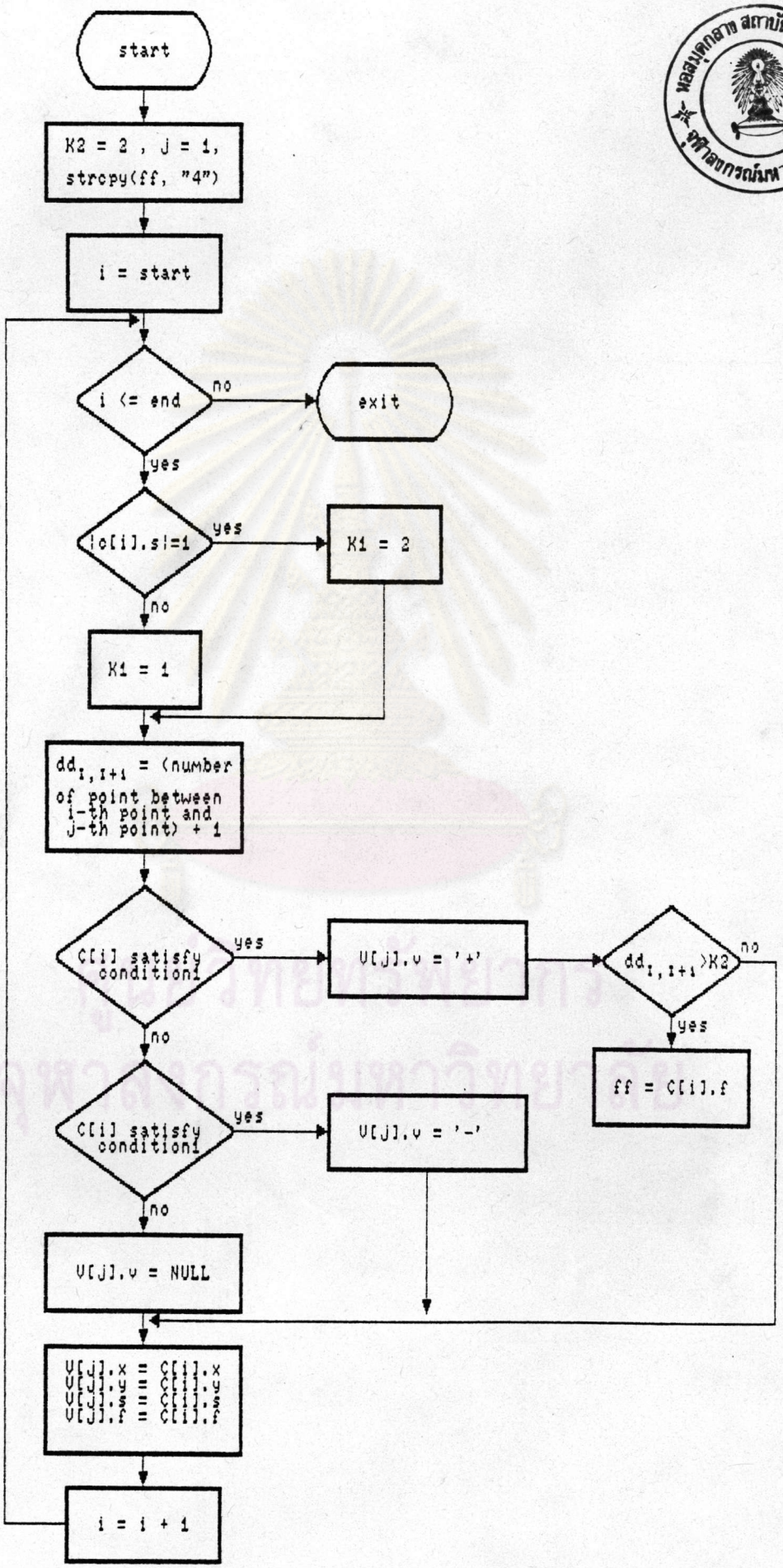
2.2 $(S_i < 0)$ และ $(S_{i-1} > 0)$ และ $(S_{i+1} > 0)$ และ $(F_i \neq FF)$ และ $(DD_{i,i+1} \geq K1)$

ดังนั้น จะได้ว่า V_j เป็นลำดับของจุดบ่งความนูน (หรือจุดบ่งความเว้า) บนเส้นแสดงขอบของอักขระ

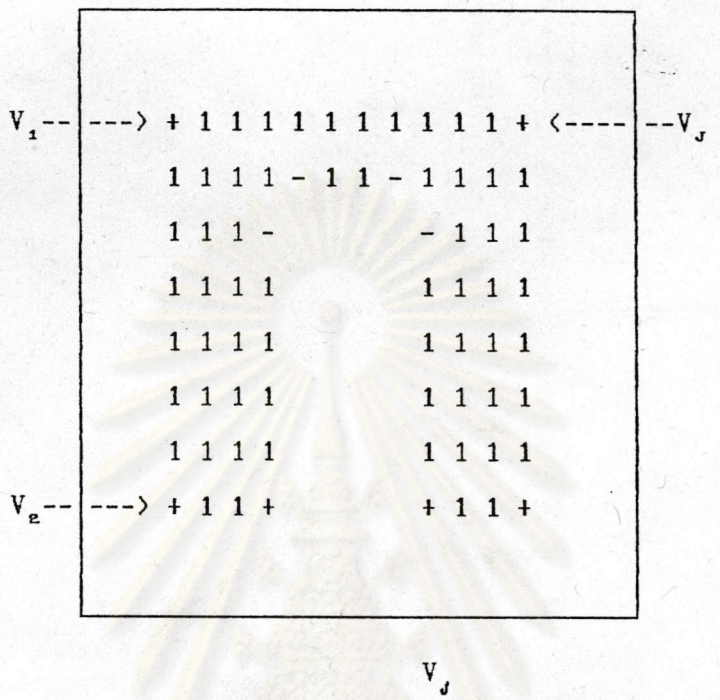
โดย $V_j = (V_{xi}, V_{yi})$: $j = 1,\ldots,J$ และ $J < I$

และ J เป็นจำนวนจุดบ่งความนูน (หรือจุดบ่งความเว้า) ทั้งหมดบนเส้นแสดงขอบของอักขระ

รูปที่ 3.20 ผังงานแสดงการกำหนดจุดบ่งความนูนและจุดบ่งความเว้า



เมตริกซ์ของอักขระที่ได้หลังจากการกำหนดจุดบ่งความนูนและจุดบ่งความเว้าตามข้อกำหนดทั้งสองแล้วจะเป็นดังรูปที่ 3.21



รูปที่ 3.21 จุดบ่งความนูนและจุดบ่งความเว้าบนเส้นแสดงขอบของอักขระ

3.4.5 การหาระยะห่างระหว่างจุดบ่งความนูนหรือจุดบ่งความเว้า

เมื่อทราบตำแหน่งของจุดบ่งความนูนและจุดบ่งความเว้าบนเส้นแสดงขอบของอักขระแล้ว ก็สามารถที่จะหาระยะห่างระหว่างจุดบ่งความนูนหรือจุดบ่งความเว้าแต่ละจุดได้โดยการคำนวณดังต่อไปนี้

สมมุติให้ จุด V_1 และ จุด V_2 เป็นจุดบ่งความนูนหรือจุดบ่งความเว้า 2 จุด ที่อยู่ถัดกันไป

และให้ จุด $V_1 = (x_1, y_1)$

จุด $V_2 = (x_2, y_2)$

และให้ DD เป็นระยะห่างระหว่างจุดบ่งความนูนหรือจุดบ่งความเว้าทั้งสองจุด

ดังนั้นระยะห่างระหว่างจุดบ่งความนูนหรือจุดบ่งความเว้า V_1 และ V_2 หาได้ ดังนี้

$$DD = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

ระยะห่าง หรือ ความยาวระหว่างจุดบ่งความนูนหรือจุดบ่งความเว้าที่คำนวณได้นี้จะถูกนำไปใช้เป็นลักษณะสำคัญในขั้นตอนของการเปรียบเทียบแบบไดนามิกโปรแกรมมิงต่อไป

3.4.6 <u>การกำหนดส่วนโค้งบนเส้นแสดงขอบของอักขระ</u>

การกำหนดส่วนโค้งบนเส้นแสดงขอบของอักขระมีจุดประสงค์ เพื่อที่จะตัดแบ่งเส้นแสดงขอบของอักขระออกเป็น ส่วนโค้งเว้า และ ส่วนโค้งนูน เพื่อใช้เป็นลักษณะสำคัญของอักขระแต่ละตัวต่อไป นอกจากนี้แล้วการกำหนดส่วนโค้งเว้าและส่วนโค้งนูนให้แก่อักขระยังทำให้ทราบว่า อักขระนี้ประกอบด้วยส่วนโค้งจำนวนกี่ส่วนโค้ง (คือจำนวนส่วนโค้งเว้า + จำนวนส่วนโค้งนูน) ซึ่งจำนวนส่วนโค้งนี้ก็ถือได้ว่า เป็นลักษณะสำคัญอีกอันหนึ่ง (นอกจากจำนวนหัวของอักขระแล้ว) ที่สามารถนำไปใช้ในการแยกประเภทของอักขระได้อย่างมีประสิทธิภาพเช่นกัน ผังงานแสดงการกำหนดส่วนโค้งบนเส้นแสดงขอบของอักขระแสดงได้ดังในรูปที่ 3.22

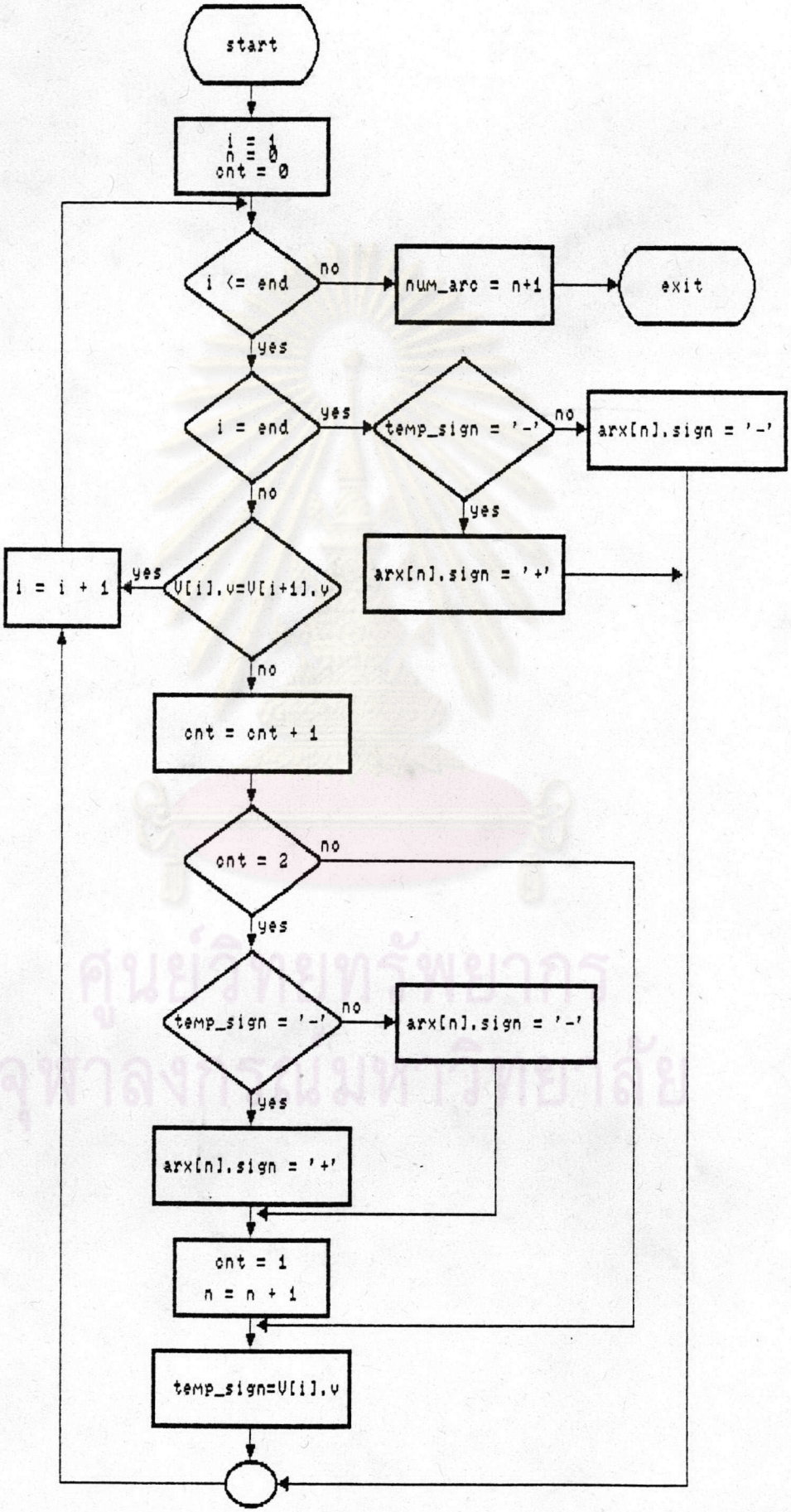
ในการตัดแบ่งเส้นขอบของอักขระออกเป็นส่วนโค้ง และกำหนดว่าส่วนโค้งใดเป็นส่วนโค้งเว้าหรือส่วนโค้งนูนนั้น จะนำคำจำกัดความของส่วนโค้งเว้าและส่วนโค้งนูนที่ได้กล่าวไว้ในบทที่ 2 มาใช้เป็นหลักในการกำหนด ดังนี้

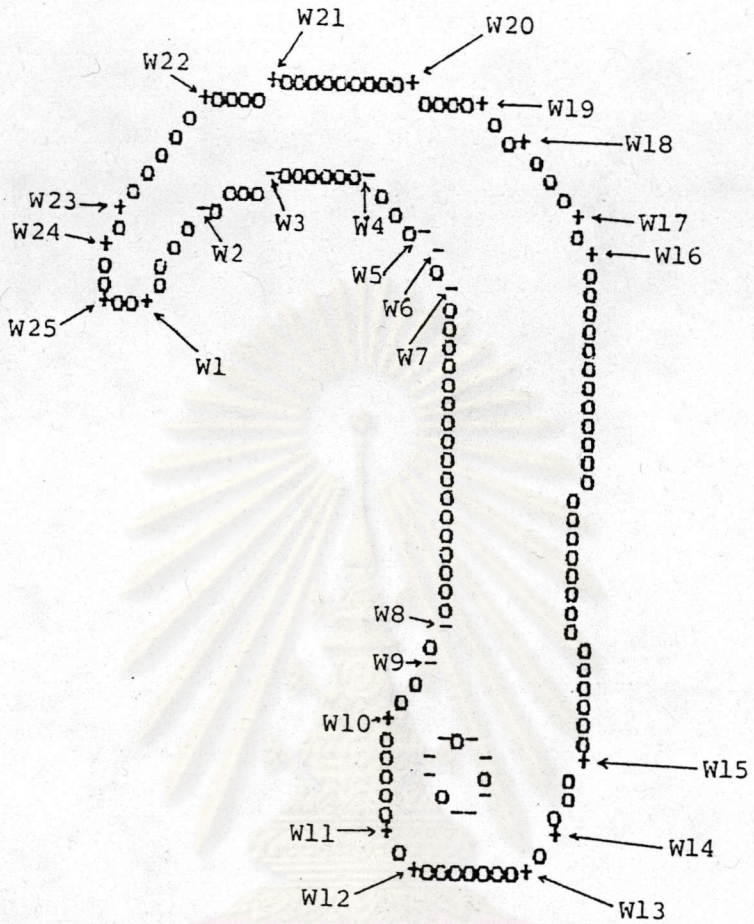
ส่วนโค้งเว้า คือ ส่วนของเส้นโค้งที่ผ่านจุดบ่งความเว้า ซึ่งอยู่ระหว่างจุดบ่งความนูน 2 จุด โดยจุดบ่งความนูนทั้งสองจุดจะเป็นจุดปลายทั้งสองของส่วนโค้งเว้านั้น

ส่วนโค้งนูน คือ ส่วนของเส้นโค้งที่ผ่านจุดบ่งความนูน ซึ่งอยู่ระหว่างจุดบ่งความเว้า 2 จุด โดยจุดบ่งความเว้าทั้งสองจุดจะเป็นจุดปลายทั้งสองของส่วนโค้งนูนนั้น

ตัวอย่างของการกำหนดจุดบ่งความนูนและจุดบ่งความเว้าให้แก่อักขระ แสดงได้ดังในรูปที่ 3.23 ซึ่งจะเห็นได้ว่า ส่วนที่ผ่านจุด W9, W10, W11, W12, W13, W14, W15, W16, W17, W18, W19, W20, W21, W22, W23, W24, W25, W1 และ W2 จะเป็นส่วนโค้งนูน และ ส่วนที่ผ่านจุด W1, W2, W3, W4, W5, W6, W7, W8, W9 และ W10 จะเป็นส่วนโค้งเว้า

รูปที่ 3.22 ผังงานแสดงการกำหนดส่วนโค้งบนเส้นแสดงขอบของอักขระ





รูปที่ 3.23 ตัวอย่างอักขระไทยที่มีการกำหนดจุดบ่งความนูนและจุดบ่งความเว้า

3.4.7 การหาจำนวนหัวของอักขระ (hole of character)

เนื่องจากอักขระไทยมีจำนวนมาก จึงได้นำลักษณะสำคัญในลักษณะภาพรวม ได้แก่ จำนวนหัวของอักขระ มาใช้เพื่อช่วยในการแยกประเภทของอักขระ ซึ่งจะช่วยทำให้สามารถลดเวลาในการรู้จำลงได้ เนื่องจากในขั้นตอนของการรู้จำนั้น ก็จะเปรียบเทียบกับเฉพาะอักขระที่อยู่ในประเภทเดียวกันเท่านั้น

อักขระในภาษาไทยมีทั้งประเภทที่ไม่มีหัวและประเภทที่มีหัว ตัวอย่างเช่น

อักขระที่ไม่มีหัว ได้แก่ ก และ ธ เพียง 2 ตัว เท่านั้น

อักขระที่มี 1 หัว เช่น ข ค ง จ ช อ เป็นต้น

อักขระที่มี 2 หัว เช่น ฆ ฉ ฌ ฒ ฌ ห เป็นต้น

ในกรณีที่เป็นอักขระที่มีหัว จะพบว่า เส้นแสดงขอบของอักขระจะมีมากกว่า 1 วง คือ วงนอกและวงใน แต่ถ้าเป็นอักขระที่ไม่มีหัวแล้ว เส้นแสดงขอบของอักขระจะมีแต่วงนอกเท่านั้น ในรูปที่ 3.24(ก) และ 3.24(ข) จะแสดงให้เห็น เส้นแสดงขอบของอักขระประเภทที่ไม่มีหัว และมีหัว ตามลำดับ

(ก) เส้นแสดงขอบของอักขระประเภทที่ไม่มีหัว (ข) เส้นแสดงขอบของอักขระประเภทที่มีหัว

รูปที่ 3.24 ตัวอย่างเส้นแสดงขอบของอักขระ

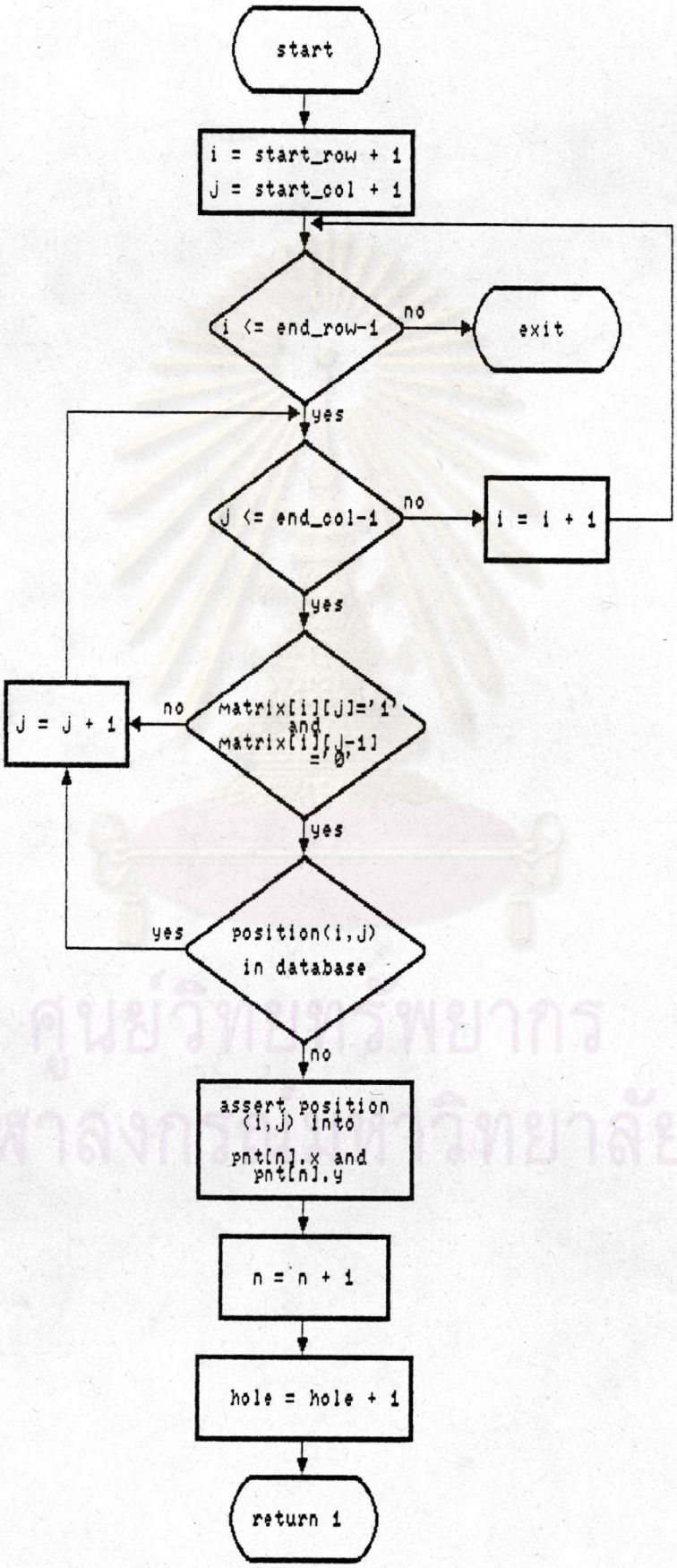
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

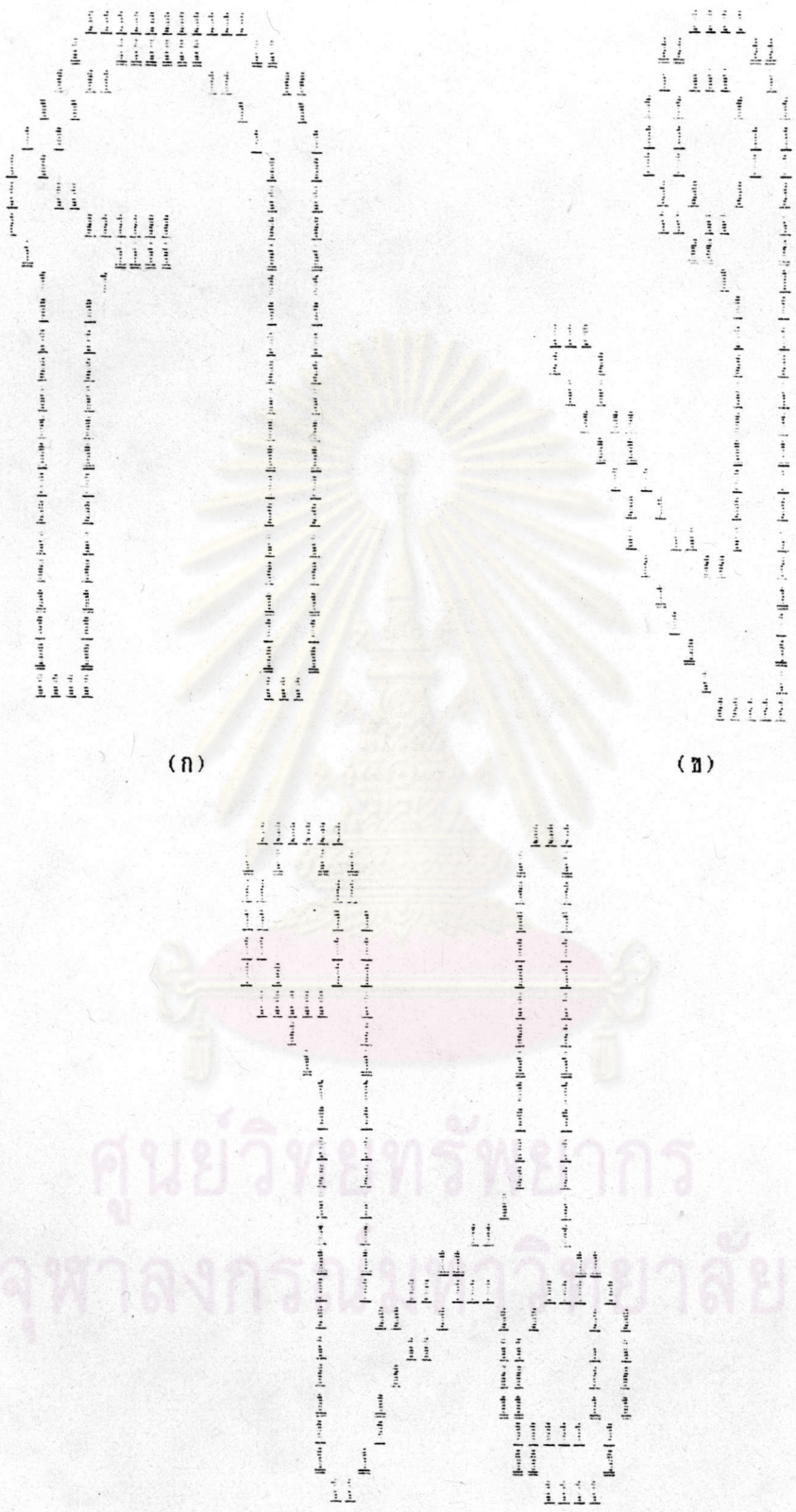
การหาจำนวนหัวของอักขระก็จะทำในลักษณะเช่นเดียวกันกับการหาเส้นแสดงขอบขอบของอักขระที่อยู่วงนอกสุด เพียงแต่ว่าในการหาจุดเริ่มต้นของเส้นแสดงขอบของอักขระในส่วนที่เป็นหัวของอักขระนี้จะต้องมีการตรวจสอบเสียก่อนว่า จุดๆ นั้นจะต้องไม่เป็นจุดที่อยู่บนเส้นแสดงขอบของอักขระที่อยู่วงนอก หรือในกรณีที่อักขระนั้นมีมากกว่า 1 หัว จุดๆ นั้นก็จะต้องไม่เป็นจุดที่อยู่บนเส้นแสดงขอบของอักขระที่เป็นส่วนหัวของอักขระอีกหัวหนึ่ง ดังนั้นจึงอาจจะมองได้อีกอย่างหนึ่งว่า เส้นที่แสดงถึงส่วนที่เป็นหัวของอักขระก็คือเส้นแสดงขอบของอักขระที่อยู่วงในนั่นเอง ผังงานแสดงการหาจำนวนหัวของอักขระแสดงได้ดังรูปที่ 3.25

ในรูปที่ 3.26(ก) จะแสดงให้เห็นถึงเส้นแสดงขอบของอักขระประเภทที่ไม่มีหัวซึ่งจะเห็นได้ว่า มีเพียงแต่เส้นแสดงขอบของอักขระที่อยู่วงนอกสุดเพียงวงเดียวเท่านั้น ส่วนในรูปที่ 3.26(ข) จะเป็นรูปที่แสดงถึงอักขระประเภทที่มีหัว 1 หัว ดังนั้นจะเห็นได้ว่า มีเส้นแสดงขอบของอักขระอยู่จำนวน 2 วง คือ วงนอกและวงใน เส้นแสดงขอบของอักขระที่อยู่วงในนี้ก็คือเส้นที่แสดงถึงส่วนที่เป็นหัวของอักขระจำนวน 1 หัวนั่นเอง สำหรับในรูปที่ 3.26(ค) จะเป็นตัวอย่างของอักขระที่มีหัว 2 หัว ซึ่งจะเห็นว่า เส้นแสดงขอบของอักขระก็จะมีจำนวนเพิ่มขึ้นเป็น 3 วง คือ วงนอก 1 วง และ วงใน 2 วง เส้นแสดงขอบของอักขระที่อยู่วงใน 2 วงนี้ก็คือส่วนที่เป็นหัวของอักขระทั้งสองหัวนั่นเอง

จำนวนหัวของอักขระที่หาได้นี้ถือได้ว่า เป็นลักษณะสำคัญอันหนึ่งซึ่งสามารถนำมาใช้ในการแยกประเภทของอักขระได้อย่างมีประสิทธิภาพ ถ้าอักขระตัวเดียวกันที่ถูกนำมากวาดตรวจด้วยแสงในแต่ละครั้งจะให้เมตริกซ์ของอักขระที่มีลักษณะเหมือนกันทุกประการ แต่ในความเป็นจริงแล้วการกวาดตรวจด้วยแสงซึ่งทำกับอักขระตัวเดิมในแต่ละครั้ง จะให้เมตริกซ์ของอักขระที่มีลักษณะแตกต่างกันเล็กน้อย เช่น ในกรณีที่บางส่วนของเส้นแสดงขอบของอักขระอยู่ชิดกันค่อนข้างมาก ผลของการกวาดตรวจด้วยแสงในแต่ละครั้งอาจได้เมตริกซ์ของอักขระที่มีลักษณะแตกต่างกันดังในรูปที่ 3.27(ก) และรูปที่ 3.27(ข) ซึ่งจะเห็นว่าในรูปที่ 3.27(ก) นั้น อักขระที่ได้จะมีจำนวนหัว 1 หัว ส่วนในรูปที่ 3.27(ข) อักขระที่ได้จะมีจำนวนหัวเป็น 2 หัว ซึ่งความผิดพลาดเช่นนี้ทำให้การใช้จำนวนหัวของอักขระในการแยกประเภทอักขระมีประสิทธิภาพลดลง

รูปที่ 3.25 ผังงานแสดงการหาจำนวนหัวของอักขระ





(ก) (ข)

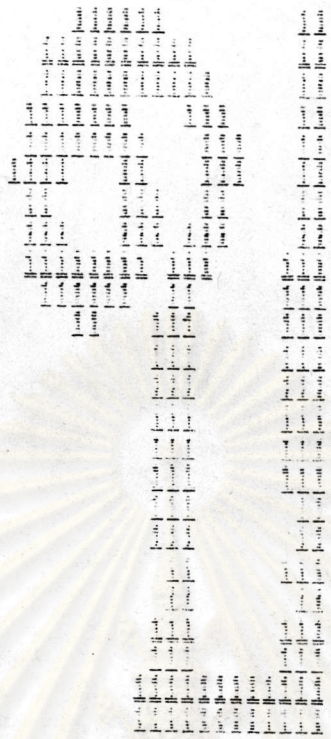
(ค)

รูปที่ 3.26(ก) เส้นแสดงขอบของอักขระประเภทที่ไม่มีหัว

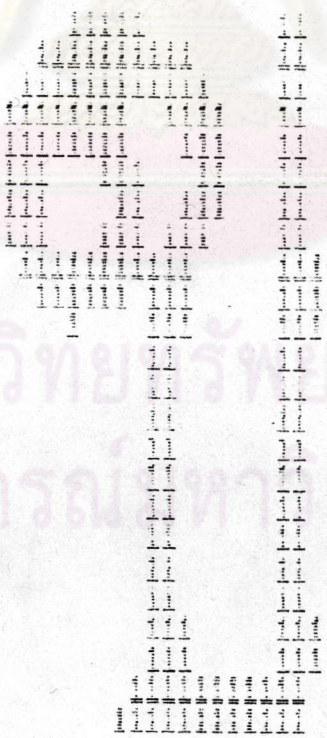
(ข) เส้นแสดงขอบของอักขระประเภทที่มีหัว 1 หัว

(ค) เส้นแสดงขอบของอักขระประเภทที่มีหัว 2 หัว

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



(ก)



(ข)

รูปที่ 3.27 ความผิดพลาดที่เกิดจากการกวาดตรวจด้วยแสงกับอักขระตัวเดิม
(ก) เมตริกซ์ของอักขระที่ได้จากการกวาดตรวจด้วยแสงครั้งแรก มีจำนวนหัว 1 หัว
(ข) เมตริกซ์ของอักขระที่ได้จากการกวาดตรวจด้วยแสงอีกครั้งหนึ่ง มีจำนวนหัว 2 หัว

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.4.8 การหาอัตราส่วนความกว้างต่อความสูงของอักขระ

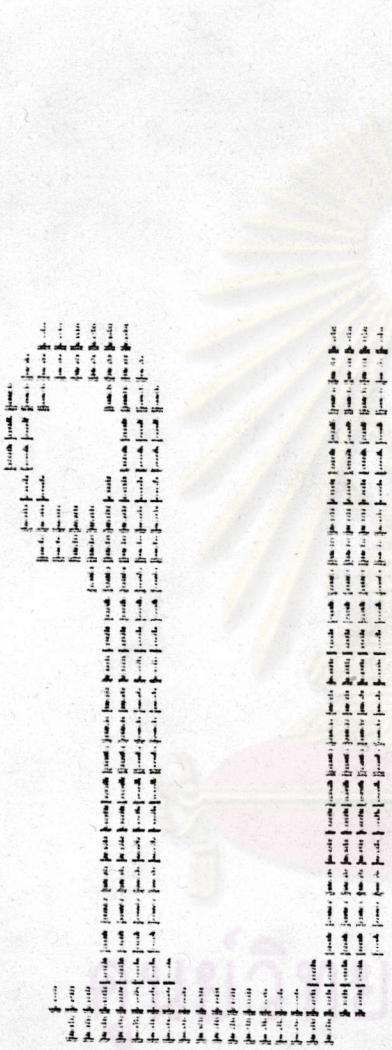
ความกว้างและความสูงของอักขระในที่นี้จะวัดจากส่วนที่กว้างที่สุดและสูงที่สุดของอักขระ โดยเมตริกซ์ของอักขระที่นำมาใช้คำนวณนี้จะเป็นเมตริกซ์ของอักขระที่ยังไม่ได้ผ่านการประมวลผลใดๆ ทั้งสิ้น ซึ่งความกว้างและความสูงของอักขระนี้จะวัดออกมาเป็นจำนวนจุด นั่นก็คือ ความกว้างของอักขระจะหาได้จากแถวที่มีจำนวนจุดมากที่สุดตามแนวแกน x และความสูงของอักขระก็จะหาได้จากสดมภ์ที่มีจำนวนจุดมากที่สุดตามแนวแกน y ดังนั้นอัตราส่วนความกว้างต่อความสูงของอักขระ (Width per Height) จะคำนวณได้ดังนี้

สมมุติให้ wph แทนอัตราส่วนความกว้างต่อความสูงของอักขระ

และให้ num_col แทนจำนวนจุดสำหรับความกว้างของอักขระ
num_row แทนจำนวนจุดสำหรับความสูงของอักขระ

ดังนั้น wph = num_col / num_row

อัตราส่วนความกว้างต่อความสูงของอักขระที่คำนวณออกมาได้นี้จะถูกนำไปใช้เป็นลักษณะสำคัญอีกอันหนึ่ง (นอกจากจำนวนหัวของอักขระ และ จำนวนส่วนโค้งของอักขระแล้ว) ในการแยกประเภทของอักขระในขั้นแรก (rough classification) เนื่องจากอักขระบางตัวเมื่อใช้จำนวนหัวและจำนวนส่วนโค้งของอักขระในการแยกประเภทแล้ว จะยังถูกจัดให้อยู่ในกลุ่มเดียวกัน เช่น บ กับ ป หรือ ผ กับ ฝ เป็นต้น แต่เมื่อนำอัตราส่วนความกว้างต่อความสูงของอักขระเข้ามาเป็นลักษณะสำคัญเพิ่มอีกลักษณะหนึ่งในการแยกประเภทแล้ว ก็จะทำให้สามารถแยก บ ออกจาก ป หรือ ผ ออกจาก ฝ ได้ ซึ่งการทำเช่นนี้นอกจากจะช่วยเพิ่มความถูกต้องในการรู้จำอักขระให้สูงขึ้นแล้ว ยังจะมีส่วนช่วยทำให้ลดจำนวนของอักขระที่จะต้องถูกนำมาเปรียบเทียบให้น้อยลง ทำให้สามารถลดเวลาในการประมวลผลเพื่อการรู้จำอักขระลงได้ ในรูปที่ 3.28(ก) และ 3.28(ข) จะเป็นตัวอย่างซึ่งเปรียบเทียบให้เห็นถึงความแตกต่างของค่าอัตราส่วนความกว้างต่อความสูงของอักขระระหว่างอักขระ บ กับ อักขระ ป ซึ่งจะเห็นว่ามีค่าแตกต่างกันมากพอที่จะสามารถใช้เป็นเกณฑ์ในการแยกประเภทอักขระได้

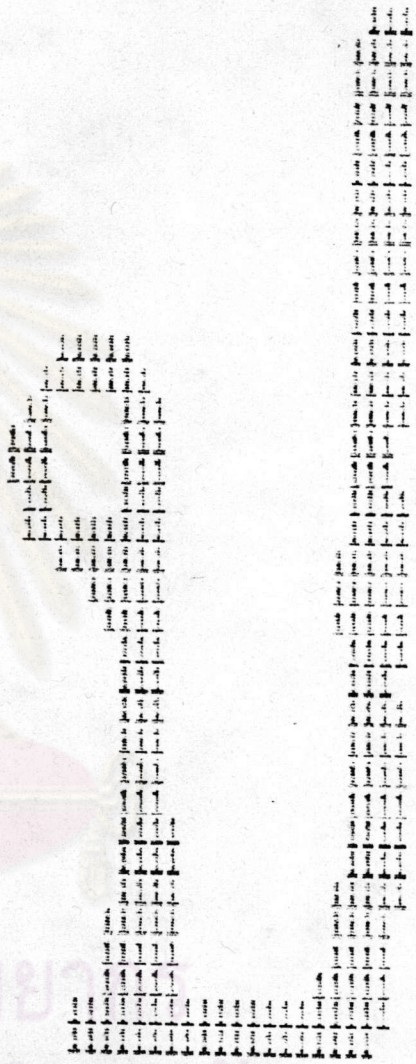


num_col = 24 จุด

num_row = 24 จุด

* wph = 24 / 24 = 1.00

(ก) อักขระ บ



num_col = 25 จุด

num_row = 35 จุด

* wph = 25 / 35 = 0.71

(ข) อักขระ ป

รูปที่ 3.28 ตัวอย่างการคำนวณค่าอัตราส่วนความกว้างต่อความสูงของอักขระ

3.5 โปรแกรมการสร้างพจนานุกรมของอักขระ

3.5.1 โครงสร้างพจนานุกรมของอักขระ

เนื่องจากในการวิจัยนี้ใช้อักขระจำนวน 3 รูปแบบ ดังนั้นในพจนานุกรมของอักขระจะประกอบด้วยอักขระทั้ง 3 รูปแบบ แต่ละรูปแบบจะประกอบด้วยอักขระ 67 ตัว เก็บเรียงต่อกันไปทั้งสามรูปแบบ โดยมีลักษณะการเก็บดังนี้

สมมุติให้ L_{ki} เป็นรหัสของอักขระตัวที่ i ของรูปแบบที่ k

และ F_{ki} เป็นลักษณะสำคัญของอักขระตัวที่ i ของรูปแบบที่ k

โดย $k = 1, \ldots, K$; K เป็นจำนวนรูปแบบทั้งหมดของอักขระ

และ $i = 1, \ldots, N$; N เป็นจำนวนอักขระทั้งหมดของแต่ละรูปแบบ

ดังนั้นสามารถจะแสดงรูปแบบการเก็บอักขระได้ดังในรูปที่ 3.29

L_{11}	F_{11}	...	L_{1N}	F_{1N}	L_{21}	F_{21}	...	L_{2N}	F_{2N}	L_{K1}	F_{K1}	...	L_{KN}	F_{KN}

รูปที่ 3.29 รูปแบบการเก็บอักขระในพจนานุกรมของอักขระ

ลักษณะสำคัญของอักขระที่เก็บอยู่ในพจนานุกรมของอักขระมีดังต่อไปนี้

1. อัตราส่วนความกว้างต่อความสูงของอักขระ
2. จำนวนหัวของอักขระ
3. จำนวนส่วนโค้งของอักขระ
4. สัญญลักษณ์ของส่วนโค้งเว้าหรือส่วนโค้งนูน
5. จำนวนจุดบ่งความเว้า (หรือจุดบ่งความนูน) ภายในส่วนโค้ง - 1
6. ความยาวระหว่างจุดบ่งความเว้าหรือจุดบ่งความนูน 2 จุด ที่อยู่ติดกันบนส่วนโค้ง

การเก็บลักษณะสำคัญของอักขระ (F_{ki}) มีรูปแบบดังนี้

สมมุติให้ wph แทนอัตราส่วนความกว้างต่อความสูงของอักขระ

hole แทนจำนวนหัวของอักขระ

arc แทนจำนวนส่วนโค้งของอักขระ

arc_sign[j] แทนสัญญลักษณ์ของส่วนโค้งเว้าหรือส่วนโค้งนูน

cnt_d[j] แทนจำนวนจุดบ่งความเว้า (หรือจุดบ่งความนูน) ภายในส่วนโค้ง - 1

เมื่อ j = 1, ... , J

และ J เป็นจำนวนส่วนโค้งของอักขระ

d[j][l] แทนความยาวระหว่างจุดบ่งความเว้าหรือจุดบ่งความนูน 2 จุด ที่อยู่ติดกันบนส่วนโค้ง

เมื่อ l = 1, ... , L

และ L = จำนวนจุดบ่งความเว้า (หรือจุดบ่งความนูน) ภายในส่วนโค้ง - 1

ดังนั้นสามารถจะแสดงการเก็บลักษณะสำคัญของอักขระ F_{ki} ได้ดังรูปที่ 3.30

wph	hole	arc	arc_sign[1]	cnt_d[1]	d[1][1]	...	d[1][L]	...

...	arc_sign[J]	cnt_d[J]	d[J][1]	...	d[J][L]

รูปที่ 3.30 รูปแบบการเก็บลักษณะสำคัญของอักขระในพจนานุกรมของอักขระ

ในรูปที่ 3.31 จะเป็นตัวอย่างแสดงให้เห็นถึงลักษณะสำคัญของอักขระ จ ดังนี้

1. wph = 0.80
2. hole = 1
3. arc = 4

4. arc_sign[j] เมื่อ j = 1, ... , 4 ดังนั้น
   arc_sign[1] = - arc_sign[2] = +
   arc_sign[3] = - arc_sign[4] = +

5. cnt_d[j] เมื่อ j = 1, ... , 4 ดังนั้น
   cnt_d[1] = 9 cnt_d[2] = 8
   cnt_d[3] = 3 cnt_d[4] = 14

6. d[j][l] เมื่อ j = 1 และ l = 1, ... , 9
   มีค่าเรียงกันดังนี้ 3, 6, 2, 2, 3, 13, 2, 2, 13

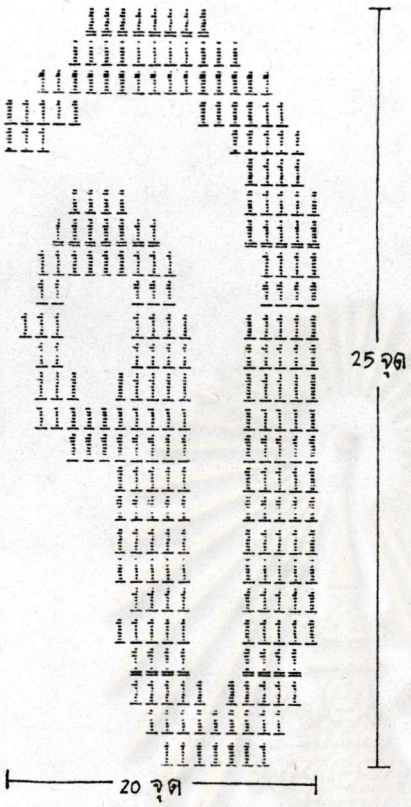
   เมื่อ j = 2 และ l = 1, ... , 8
   มีค่าเรียงกันดังนี้ 13, 1, 2, 3, 2, 5, 2, 2

   เมื่อ j = 3 และ l = 1, ... , 3
   มีค่าเรียงกันดังนี้ 2, 1, 7

   เมื่อ j = 4 และ l = 1, ... , 14
   มีค่าเรียงกันดังนี้ 7, 2, 6, 2, 2, 14, 2, 2, 4, 7,
   5, 1, 2, 3

ดังนั้นรูปแบบการเก็บอักขระ จ จะเป็นดังในรูปที่ 3.32

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

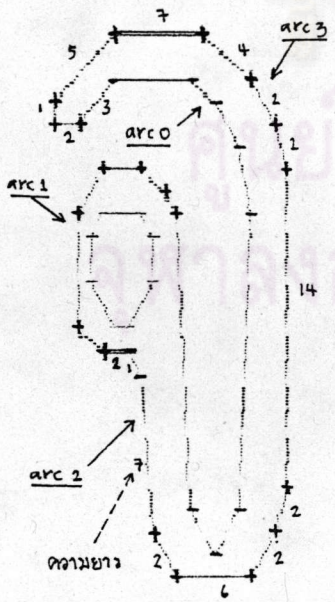


num_col = 20 จุด

num_row = 25 จุด

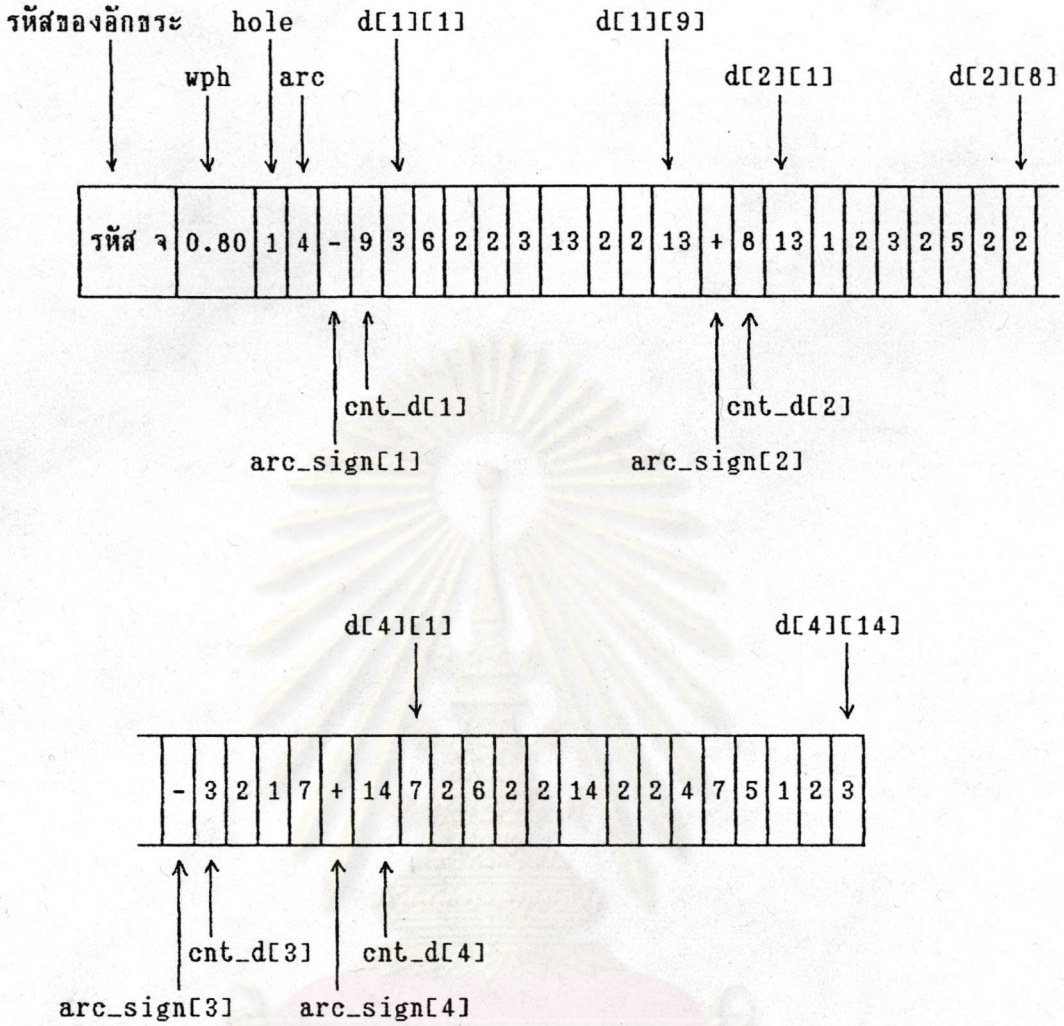
* wph = 20 / 25 = 0.80

- Hole = 1, Arc = 4 -




```
* Distance of each arc *
arc
 0: - : 3  6  2  2  3 13  2  2 13
 1: + : 13 1  2  3  2  5  2  2
 2: - : 2  1  7
 3: + : 7  2  6  2  2 14  2  2  4  7  5  1  2  3
```

รูปที่ 3.31 ตัวอย่างแสดงลักษณะสำคัญของอักขระ จ



รูปที่ 3.32 รูปแบบการเก็บอักขระ จ

3.5.2 การสร้างพจนานุกรมของอักขระ

การสร้างพจนานุกรมของอักขระแบ่งเป็น 2 ขั้นตอนย่อย คือ

3.5.2.1 การเก็บอักขระต้นแบบ

เนื่องจากในการวิจัยนี้ใช้รูปแบบของอักขระจำนวน 3 รูปแบบ และอักขระแต่ละรูปแบบจะนำไปกวาดตรวจด้วยแสงจำนวน 5 ครั้ง แล้วจึงได้นำผลของการกวาดตรวจด้วยแสงจำนวน 1 ครั้ง ใน 5 ครั้ง มาใช้เป็นอักขระต้นแบบสำหรับแต่ละรูปแบบ ดังนั้นอักขระต้นแบบในการวิจัยนี้จึงมีทั้งหมด 3 รูปแบบของอักขระ

การสร้างพจนานุกรมของอักขระในขั้นแรกนี้ จะได้นำเมตริกซ์ของอักขระที่ได้จากโปรแกรมการตัดอักขระโดยอัตโนมัติมาประมวลผล โดยใช้โปรแกรมการวิเคราะห์ลักษณะสำคัญของอักขระเพื่อหาลักษณะสำคัญของอักขระออกมา จากนั้นจึงเก็บลักษณะสำคัญของอักขระที่ได้มาเหล่านี้ พร้อมทั้งรหัสของอักขระตัวนั้นลงในพจนานุกรมของอักขระ โดยรูปแบบการเก็บเป็นดังที่ได้กล่าวไว้ในหัวข้อโครงสร้างพจนานุกรมของอักขระ ผังงานแสดงการเก็บอักขระต้นแบบในขั้นแรกนี้จะได้แสดงไว้ในรูปที่ 3.33

3.5.2.2 การเรียนรู้ (learning)

ก. กรณีที่เป็นอักขระรูปแบบใหม่ซึ่งไม่อยู่ในพจนานุกรมของอักขระ

ในกรณีที่อักขระรูปแบบใหม่มีลักษณะคล้ายคลึง กับอักขระที่มีเก็บอยู่แล้วในพจนานุกรมของอักขระ ก็จะไม่ทำให้ความถูกต้องในการรู้จำลดลงมากนัก แต่สำหรับในกรณีที่อักขระรูปแบบใหม่มีลักษณะ ซึ่งแตกต่างจากลักษณะของอักขระที่เก็บไว้ในพจนานุกรมของอักขระมาก ก็อาจจะมีผลทำให้ความถูกต้องในการรู้จำอักขระลดลงได้มาก ดังนั้นจึงควรให้เครื่องคอมพิวเตอร์ได้มีการเรียนรู้รูปแบบใหม่ของอักขระด้วย

การจัดการให้เครื่องคอมพิวเตอร์ ได้มีการเรียนรู้รูปแบบที่ต้องการก่อนจะมีส่วนช่วยทำให้ผลในการรู้จำถูกต้องมากยิ่งขึ้น เนื่องจากเมื่อเครื่องคอมพิวเตอร์ได้มีการเรียนรู้รูปแบบที่ต้องการแล้ว เครื่องก็จะจัดการเก็บบันทึกรูปแบบของอักขระรูปแบบใหม่เพิ่มเข้าไปในพจนานุกรมของอักขระด้วย ทำให้เมื่อนำอักขระรูปแบบใหม่มาดำเนินการรู้จำเครื่องก็สามารถจะค้นหาอักขระที่มีรูปแบบใหม่ในพจนานุกรมของอักขระพบได้ ทำให้ความถูกต้องในการรู้จำอักขระรูปแบบใหม่เพิ่มมากขึ้นกว่าในกรณีที่ไม่ให้มีการเรียนรู้อักขระรูปแบบใหม่เลย

ข. กรณีที่เป็นอักขระรูปแบบเดิมซึ่งมีอยู่ในพจนานุกรมของอักขระ

กรณีที่รูปแบบของอักขระเป็นรูปแบบ ซึ่งมีเก็บอยู่แล้วในพจนานุกรมของอักขระนั้น การให้เครื่องคอมพิวเตอร์ได้เรียนรู้รูปแบบเดิมที่ได้จากการกวาดตรวจด้วยแสงหลายๆ ครั้งก็จะทำให้ผลของการรู้จำมีความถูกต้องสูงขึ้นได้มาก เนื่องจากในการกวาดตรวจด้วยแสงแต่ละครั้งนั้น ข้อมูลของอักขระตัวเดิมที่ได้จะมีลักษณะแตกต่างกันออกไปเสมอ กล่าวคือ ในการกวาดตรวจด้วยแสงกับอักขระตัวเดิมนั้น ผลที่ได้ในแต่ละครั้งของการกวาดตรวจอาจได้อักขระที่มีลักษณะแตกต่างกันออกไปมาก จนทำให้เหมือนกับว่าเป็นอักขระรูปแบบใหม่ไปเลยก็ได้

สาเหตุซึ่งทำให้การกวาดตรวจด้วยแสงในแต่ละครั้ง กับอักขระตัวเดิมได้ผลแตกต่างกันอาจเนื่องมาจากสาเหตุ ดังต่อไปนี้

- จากตัวอุปกรณ์เครื่องกวาดตรวจด้วยแสงเอง

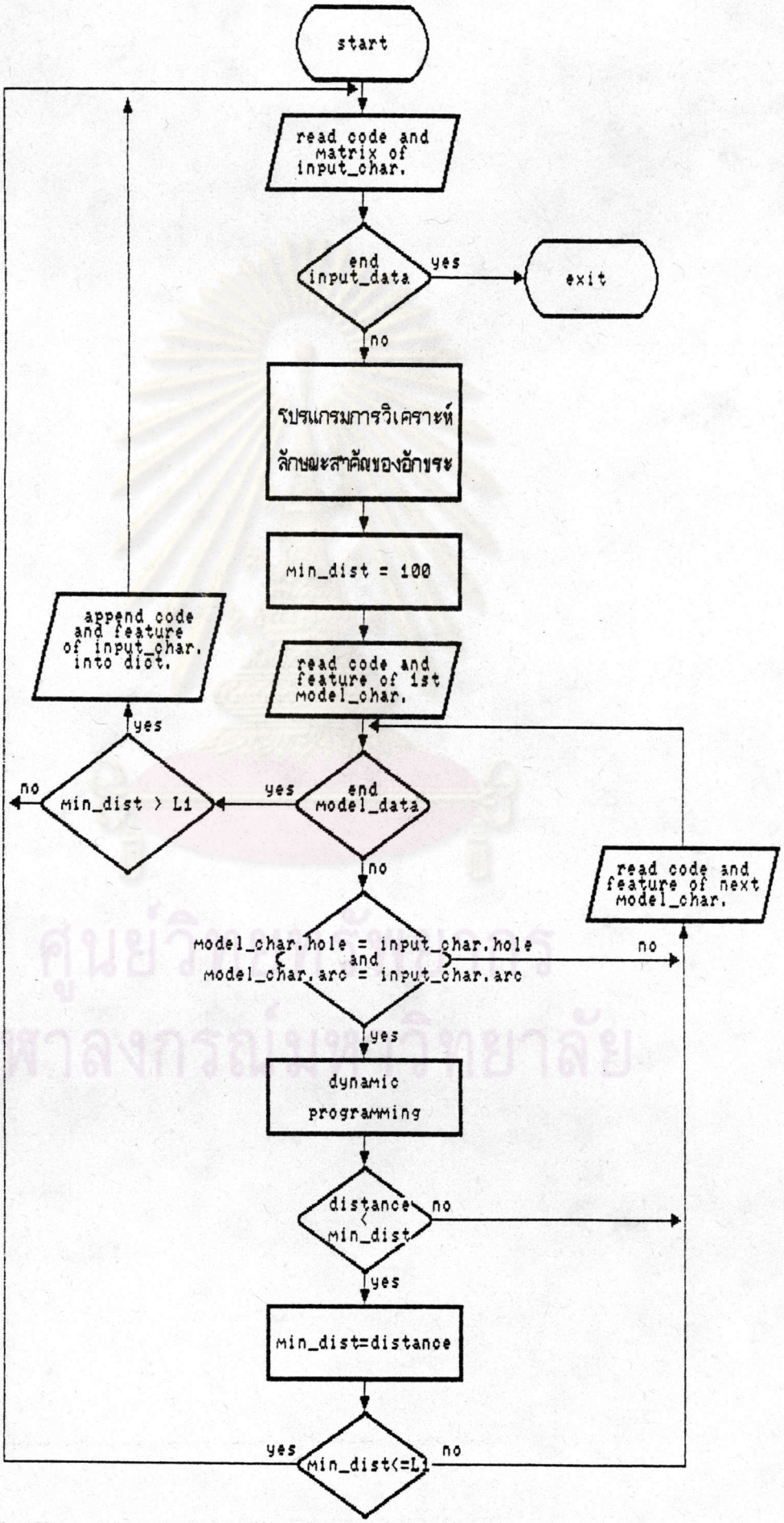
- จากลักษณะการวางกระดาษของผู้ใช้ ซึ่งในการวางกระดาษแต่ละครั้งจะเป็นไปไม่ได้ที่จะวางได้เหมือนเดิมทุกครั้ง หรือ แม้จะไม่มีการยกกระดาษขึ้นเลยแล้วกวาดตรวจด้วยแสงหลายครั้ง ลักษณะข้อมูลของอักขระตัวเดิมที่ได้ก็ยังมีลักษณะแตกต่างกันอยู่เพียงแต่จะไม่แตกต่างมากเท่ากับกรณีที่มีการยกกระดาษแล้ววางใหม่เท่านั้น

- จากเวลา คือ ถ้าพึ่งจะเริ่มเปิดเครื่องกวาดตรวจด้วยแสงใหม่ๆ แล้วกวาดตรวจด้วยแสงในทันที ลักษณะข้อมูลของอักขระตัวเดิมที่ได้ก็จะมีลักษณะแตกต่างจากในกรณีที่เปิดเครื่องทิ้งไว้สักครู่แล้วจึงค่อยเริ่มกวาดตรวจด้วยแสง

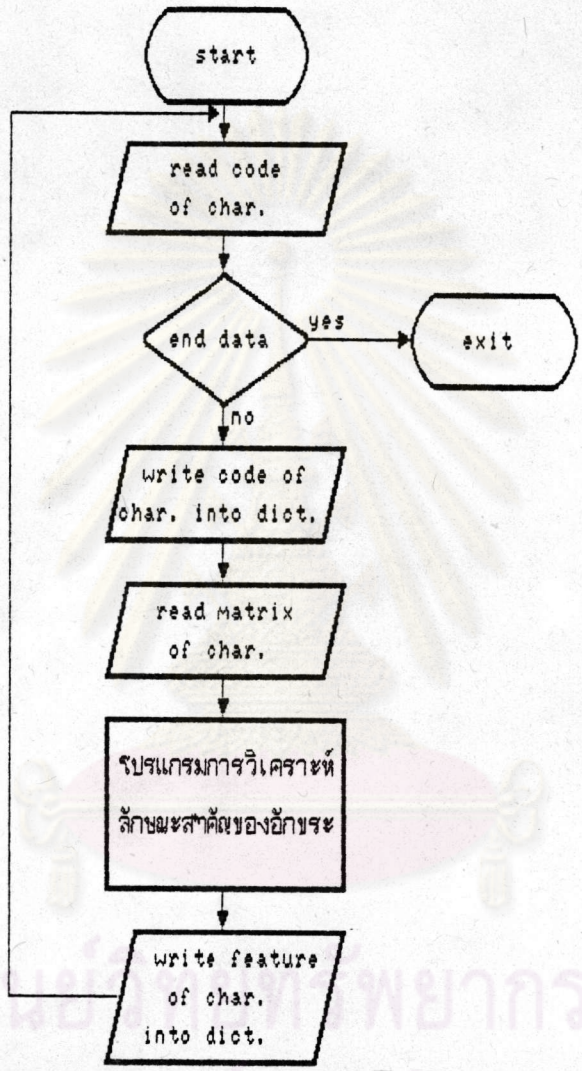
การให้เครื่องคอมพิวเตอร์เรียนรู้รูปแบบของอักขระไม่ว่ารูปแบบนั้นจะมีอยู่ในพจนานุกรมของอักขระแล้วหรือไม่ก็ตาม ทำโดยการนำข้อมูลที่ได้จากการกวาดตรวจด้วยแสงมาผ่านการประมวลผลของโปรแกรมการตัดอักขระโดยอัตโนมัติ และ โปรแกรมการวิเคราะห์ลักษณะสำคัญของอักขระ จากนั้นจึงนำลักษณะสำคัญของอักขระที่ได้รับเข้ามาใหม่ไปเปรียบเทียบกับลักษณะสำคัญของอักขระที่เก็บไว้ในพจนานุกรมของอักขระ โดยใช้วิธีการเปรียบเทียบแบบไดนามิกโปรแกรมมิง ถ้าค่าความแตกต่างที่คำนวณได้ระหว่างอักขระที่รับเข้ามาใหม่กับอักขระต้นแบบมีค่ามากเกินกว่าค่าที่กำหนดไว้ค่าหนึ่ง แสดงว่าอักขระที่รับเข้ามาใหม่นี้มีลักษณะแตกต่างจากลักษณะของอักขระที่เก็บไว้ในพจนานุกรมของอักขระมาก ดังนั้นก็จะนำลักษณะสำคัญของอักขระที่รับเข้ามาใหม่นั้นเก็บเพิ่มเข้าไว้ในพจนานุกรมของอักขระ พร้อมทั้งกำหนดรหัสของอักขระนั้นไว้ให้ด้วย

การที่ให้เครื่องคอมพิวเตอร์ได้เรียนรู้อักขระรูปแบบใหม่ หรือ อักขระรูปแบบเดิมที่ได้จากการกวาดตรวจด้วยแสงหลายครั้ง จะมีส่วนช่วยทำให้ผลของความถูกต้องในการรู้จำสูงขึ้นได้มาก แต่ถ้าข้อกำหนดที่ใช้ในการวัดความแตกต่างไม่เหมาะสมก็อาจทำให้มีจำนวนอักขระในพจนานุกรมของอักขระมากเกินไปทำให้ใช้เวลาในการรู้จำอักขระมากยิ่งขึ้น โดยที่ความถูกต้องในการรู้จำไม่ได้เพิ่มขึ้นในอัตราส่วนเดียวกัน ผังงานแสดงการเรียนรู้อักขระเป็นดังในรูปที่ 3.34

รูปที่ 3.33 ผังงานแสดงการเก็บอักขระต้นแบบสำหรับเป็นพจนานุกรมของอักขระ



รูปที่ 3.34 ผังงานแสดงการเรียนรู้อักขระ



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

3.6 โปรแกรมการรู้จำอักขระ

โปรแกรมนี้จะอ่านเมตริกซ์ของอักขระเข้ามา และ หาลักษณะสำคัญของอักขระ เพื่อนำไปใช้ ในการเปรียบเทียบกับอักขระในพจนานุกรมต้นแบบของอักขระที่สร้างไว้ในขั้นตอนของการเรียนรู้ แล้ววิเคราะห์ออกมาว่ามีความใกล้เคียงกับรูปแบบของอักขระใดมากที่สุด ก็จะให้ผลลัพธ์ออกมาเป็นรหัสของอักขระที่รู้จำได้

3.6.1 การรับอักขระเข้าสู่โปรแกรม

อักขระที่เข้าสู่โปรแกรมนี้จะได้มาจาก แฟ้มข้อมูลที่เป็นผลลัพธ์จากการประมวลผลของโปรแกรมการตัดอักขระโดยอัตโนมัติ ซึ่งแฟ้มข้อมูลนี้จะเก็บเมตริกซ์ของอักขระทั้งหมดที่ต้องการรู้จำไว้ เมตริกซ์ของอักขระจะถูกอ่านเข้ามาทีละเมตริกซ์ แล้วจึงนำเมตริกซ์ของอักขระนี้ไปผ่านการประมวลผลด้วยโปรแกรมการวิเคราะห์ลักษณะสำคัญของอักขระ เพื่อหาลักษณะสำคัญของอักขระที่ต้องการรู้จำ สำหรับนำไปเปรียบเทียบกับลักษณะสำคัญของอักขระที่เก็บไว้ในพจนานุกรมของอักขระ โดยหลังจากการเปรียบเทียบอัตราส่วนความกว้างต่อความสูง จำนวนหัว และจำนวนส่วนโค้งของอักขระที่รับเข้ามากับอักขระต้นแบบแล้ว จึงจะเข้าสู่ขั้นตอนของการเปรียบเทียบแบบไดนามิกโปรแกรมมิงต่อไป

3.6.2 การเปรียบเทียบอักขระโดยวิธีการไดนามิกโปรแกรมมิง

เมื่อทราบลักษณะสำคัญของอักขระที่ต้องการรู้จำแล้ว ก็สามารถนำลักษณะสำคัญนี้ไปใช้ในการเปรียบเทียบกับอักขระต้นแบบที่เก็บอยู่ในพจนานุกรมของอักขระ เพื่อที่จะรู้จำให้ได้ว่า อักขระที่อ่านเข้ามานี้ คืออักขระใด ซึ่งก่อนที่จะเข้าสู่วิธีการเปรียบเทียบแบบไดนามิกโปรแกรมมิงนี้ ลักษณะสำคัญของอักขระได้แก่ อัตราส่วนความกว้างต่อความสูงของอักขระ จำนวนหัวของอักขระ และ จำนวนส่วนโค้งของอักขระจะถูกนำมาใช้ เพื่อการแยกประเภทของอักขระในขั้นแรกก่อน การแยกประเภทของอักขระในขั้นแรกนี้ นอกจากจะช่วยทำให้สามารถลดจำนวนอักขระที่จะต้องถูกนำไปเปรียบเทียบในขั้นตอนของไดนามิกโปรแกรมมิงลงได้แล้ว ยังจะช่วยทำให้ลดเวลาในขั้นตอนของการรู้จำอักขระลงได้อีกด้วย

3.6.2.1 ลักษณะสำคัญที่จะนำไปใช้ในขั้นตอนของการเปรียบเทียบ

ในรูปที่ 3.31 จะเห็นว่า อักขระ จ ประกอบด้วย 4 ส่วนโค้ง และในแต่ละส่วนโค้งก็จะประกอบไปด้วยจุดเปลี่ยนทิศทางที่มีเครื่องหมายกำกับอยู่ ซึ่งความยาว

ระหว่างจุดบ่งความนูนหรือจุดบ่งความเว้า 2 จุด ที่อยู่ติดกันบนส่วนโค้งนี้ จะถูกนำมาใช้เป็นลักษณะสำคัญอีกอย่างหนึ่งของอักขระในขั้นตอนของการเปรียบเทียบแบบไดนามิกโปรแกรมมิง

เนื่องจากได้นำ วิธีการเปรียบเทียบแบบไดนามิกโปรแกรมมิงมาประยุกต์ใช้กับการรู้จำอักขระในระบบออฟไลน์ ดังนั้นในการวิจัยนี้จึงได้กำหนดให้ส่วนโค้งแต่ละส่วนโค้งเป็นเวกเตอร์ที่แสดงลักษณะของรูปแบบ ดังนี้

ส่วนโค้ง L แทน เวกเตอร์ของรูปแบบของอักขระที่ต้องการรู้จำ
ส่วนโค้ง M แทน เวกเตอร์ของรูปแบบของอักขระต้นแบบ

ดังนั้นสามารถเขียนส่วนโค้ง L และ M ในรูปของเวกเตอร์ได้ ดังนี้

ส่วนโค้ง $L = l_1, l_2, \ldots, l_i, \ldots, l_I$
ส่วนโค้ง $M = m_1, m_2, \ldots, m_j, \ldots, m_J$

โดย l_i และ m_j จะเป็นลำดับของฟีเชอร์ของส่วนโค้ง ซึ่งในที่นี้ก็คือ ความยาวระหว่างจุดบ่งความนูนหรือจุดบ่งความเว้า 2 จุด ที่อยู่ติดกันของส่วนโค้ง ซึ่งความยาวนี้เป็นลักษณะสำคัญของอักขระ ที่จะถูกนำมาใช้ในขั้นตอนของการเปรียบเทียบแบบไดนามิกโปรแกรมมิง ดังนั้นการหาค่าความแตกต่างระหว่างเวกเตอร์ของรูปแบบทั้งสองเวกเตอร์ คือ ส่วนโค้ง L และ ส่วนโค้ง M ก็คือ การหาค่าความแตกต่างระหว่างส่วนโค้งของอักขระที่ต้องการรู้จำกับอักขระต้นแบบ ซึ่งการหาค่าความแตกต่างนี้จะเป็นการทำเพื่อหาส่วนโค้งคู่ที่มีความคล้ายกันมากที่สุดระหว่างอักขระที่ต้องการรู้จำกับอักขระต้นแบบ สำหรับใช้เป็นส่วนโค้งเริ่มต้นในการหาค่าความแตกต่างระหว่างอักขระที่ต้องการรู้จำกับอักขระต้นแบบต่อไป

ด้วยวิธีการของไดนามิกโปรแกรมมิง สามารถหาความแตกต่างระหว่างเวกเตอร์ l_i และ m_j ได้ดังนี้

ให้ $d(i,j)$ แทนความแตกต่างระหว่างเวกเตอร์ l_i และ m_j
ดังนั้น $d(i,j) = | l_i - m_j |$

เมื่อหาความแตกต่างระหว่างเวกเตอร์ l_i และ m_j ได้แล้ว ก็จะนำค่าความแตกต่างนี้มาใช้ในการคำนวณหาความแตกต่างระหว่างส่วนโค้ง L และ M ได้ดังนี้

ให้ D(L,M) แทน ความแตกต่างระหว่างส่วนโค้ง L และ M

ดังนั้น D(L,M) = G(I,J) / N

เมื่อ N = I + J ในกรณีของแกนเวลาสมมาตร

N = I ในกรณีของแกนเวลาไม่สมมาตร

3.6.2.2 การหาส่วนโค้งคู่ที่มีความคล้ายกันมากที่สุด

ในการหาค่าความแตกต่างระหว่างส่วนโค้ง L และ M นี้ จะหาเฉพาะคู่ของส่วนโค้งที่มีลักษณะเดียวกัน คือ เป็นส่วนโค้งเว้าหรือส่วนโค้งนูนเหมือนกัน ดังนี้

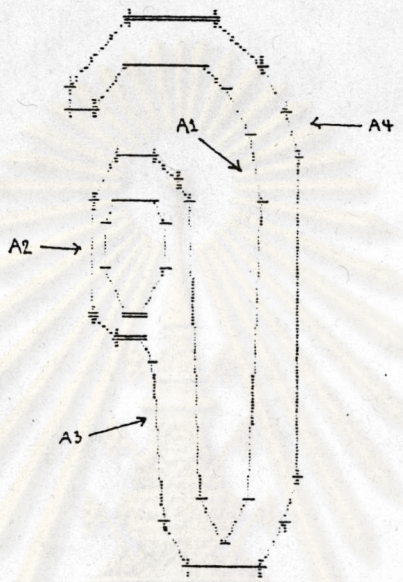
ให้อักขระ จ1 มาจากการกวาดตรวจด้วยแสงครั้งที่ 1 (อักขระต้นแบบ) และ ให้อักขระ จ2 มาจากการกวาดตรวจด้วยแสงครั้งที่ 2 (อักขระที่ต้องการรู้จำ) ดังแสดงในรูปที่ 3.35 ซึ่งจากรูปนี้จะเห็นได้ว่า อักขระ จ1 และ จ2 ประกอบด้วยส่วนโค้งย่อยจำนวน 4 ส่วนโค้ง โดยแต่ละส่วนโค้งของอักขระจะมีลักษณะของความโค้งเว้าที่แตกต่างกัน ดังแสดงในตารางที่ 3.1

ตารางที่ 3.1 ลักษณะของความโค้งเว้าของอักขระ จ1 และ อักขระ จ2

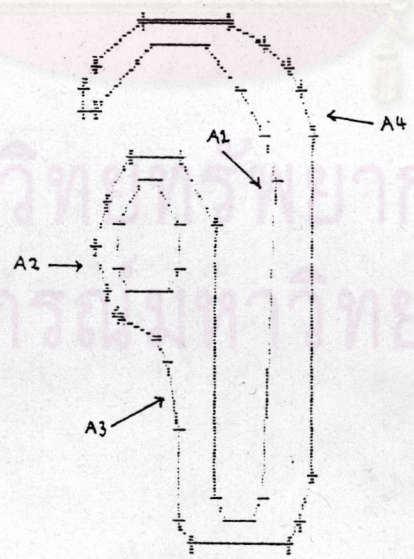
อักขระ	จ1	จ2
ส่วนโค้งที่ 1	ส่วนโค้งเว้า	ส่วนโค้งเว้า
ส่วนโค้งที่ 2	ส่วนโค้งนูน	ส่วนโค้งนูน
ส่วนโค้งที่ 3	ส่วนโค้งเว้า	ส่วนโค้งเว้า
ส่วนโค้งที่ 4	ส่วนโค้งนูน	ส่วนโค้งนูน

ดังนั้นในการหาส่วนโค้งคู่ที่มีความแตกต่างกันน้อยที่สุด ก็จะทำโดยการหาค่าความแตกต่างระหว่างทุกคู่ส่วนโค้งของอักขระทั้งสองนั้น (อักขระ จ1 และ จ2)

เพื่อที่จะหาให้ได้ว่าส่วนโค้งคู่ใดเป็นส่วนโค้งคู่ที่มีความคล้ายกันมากที่สุด ตารางที่ 3.2 จะแสดงให้เห็นถึง ค่าความแตกต่างที่ได้จากการเปรียบเทียบเพื่อหาส่วนโค้งคู่ที่คล้ายกันมากที่สุดระหว่างอักขระ จ1 และ จ2 โดยใช้สมการไดนามิกโปรแกรมมิงแบบไม่สมมาตรที่ใช้ค่าความชันเป็น 1 (สมการที่ T.6 ในตารางที่ 2.1)



(ก) อักขระ จ1



(ข) อักขระ จ2

รูปที่ 3.35 ตัวอย่างแสดงลักษณะสำคัญของอักขระ จ1 และ อักขระ จ2

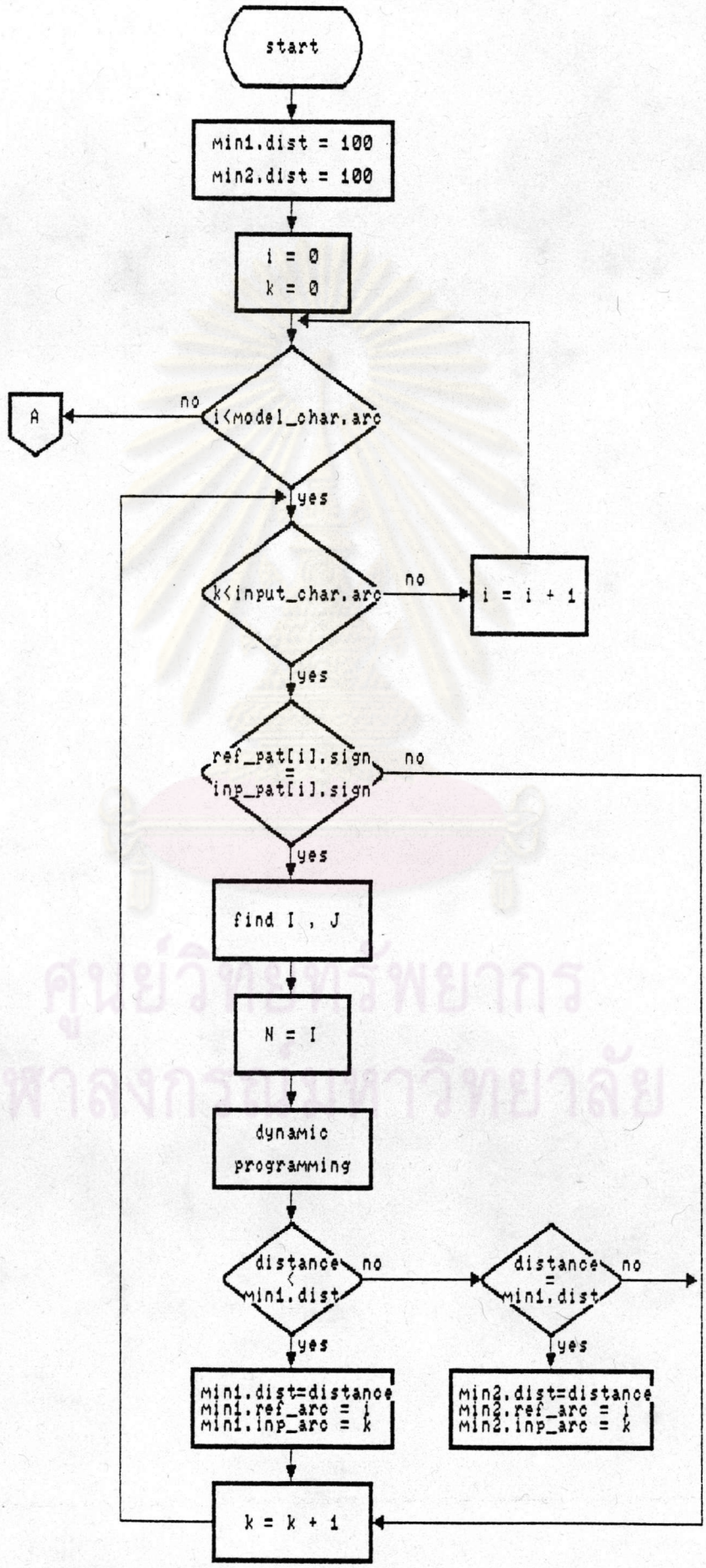
ตารางที่ 3.2 การเปรียบเทียบเพื่อหาส่วนโค้งคู่ที่คล้ายกันมากที่สุดระหว่างอักขระ จ1 และ จ2

D(AL,AM)	ค่าที่ได้
D(A1,A1)	1.222
D(A1,A3)	35.000
D(A2,A2)	1.250
D(A2,A4)	2.857
D(A3,A1)	12.222
D(A3,A3)	1.833
D(A4,A2)	13.188
D(A4,A4)	1.179

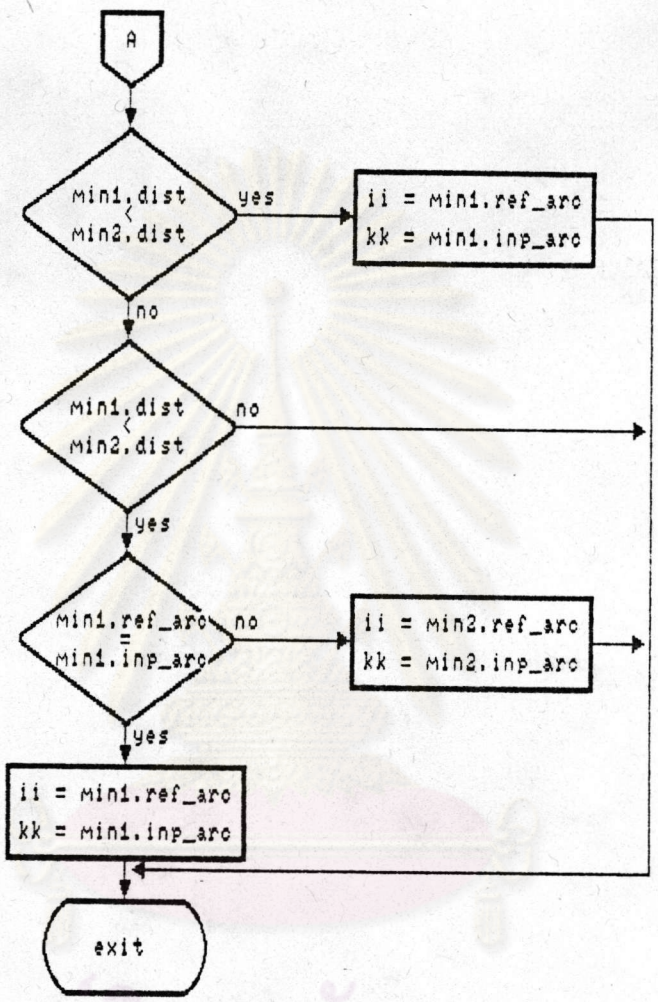
เมื่อ D(AL,AM) หมายถึง ค่าความแตกต่างระหว่างส่วนโค้ง AL ของอักขระที่ต้องการรู้จำ (อักขระ จ2) กับ ส่วนโค้ง AM ของอักขระต้นแบบ (อักขระ จ1) ซึ่งเมื่อพิจารณาค่าความแตกต่างที่ได้แล้วจะเห็นได้ว่า คู่ของส่วนโค้งที่คล้ายกันมากที่สุด ได้แก่ ส่วนโค้ง A4 ของอักขระ จ2 กับ ส่วนโค้ง A4 ของอักขระ จ1 เนื่องจากเป็นส่วนโค้งคู่ที่มีค่าความแตกต่างระหว่างส่วนโค้งน้อยที่สุด คือ มีค่าเท่ากับ 1.179

เมื่อได้คู่ส่วนโค้งที่คล้ายกันมากที่สุด ระหว่างอักขระที่ต้องการรู้จำกับอักขระต้นแบบแล้ว ก็จะใช้คู่ส่วนโค้งนี้เป็นคู่ส่วนโค้งเริ่มต้นสำหรับใช้ในการหาค่าความแตกต่างระหว่างอักขระที่ต้องการรู้จำกับอักขระต้นแบบต่อไป ผังงานแสดงการหาส่วนโค้งคู่ที่มีความคล้ายกันมากที่สุดเป็นดังรูปที่ 3.36

รูปที่ 3.36 ผังงานแสดงการหาส่วนโค้งคู่ที่มีความคล้ายกันมากที่สุด



รูปที่ 3.36 (ต่อ)



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.6.2.3 การหาอักขระต้นแบบที่มีความคล้ายกับอักขระที่ต้องการรู้จำมากที่สุด

เมื่อได้ส่วนโค้งคู่ที่มีความคล้ายกันมากที่สุดระหว่าง อักขระที่ต้องการรู้จำกับอักขระต้นแบบแล้ว ก็จะใช้คู่ส่วนโค้งนี้เป็นคู่ส่วนโค้งเริ่มต้นสำหรับการหาค่าความแตกต่างระหว่างส่วนโค้งคู่ถัดไปจนครบหมดทุกส่วนโค้ง จากนั้นก็จะนำค่าความแตกต่างของทุกคู่ของส่วนโค้งที่ได้นี้มาหาผลรวม แล้วหารด้วยจำนวนส่วนโค้งอีกครั้งหนึ่ง ซึ่งผลลัพธ์ที่ได้ในครั้งสุดท้ายนี้ก็คือ ค่าความแตกต่างระหว่างอักขระที่ต้องการรู้จำกับอักขระต้นแบบ ผังงานแสดงการหาอักขระต้นแบบที่มีความคล้ายกับอักขระที่ต้องการรู้จำมากที่สุดเป็นดังรูปที่ 3.37

ให้ $DC_{II.MM}$ แทนความแตกต่างระหว่างอักขระที่ต้องการรู้จำกับอักขระต้นแบบ

Z แทนจำนวนส่วนโค้ง

D(L,M) แทนความแตกต่างระหว่างส่วนโค้ง L และ M

ดังนั้น $DC_{II.MM} = \sum_{L,M} D(L,M) / Z$

จากตารางที่ 3.2 จะได้ว่า คู่ของส่วนโค้งที่คล้ายกันมากที่สุดได้แก่ ส่วนโค้ง A4 ของอักขระ จ2 กับ ส่วนโค้ง A4 ของอักขระ จ1 ดังนั้นการหาผลรวมของค่าความแตกต่างของคู่ส่วนโค้งจะมีลำดับดังแสดงในตารางที่ 3.3 ซึ่งค่าที่ได้ในตารางนี้จะแสดงให้เห็นถึง ค่าความแตกต่างที่ได้ระหว่างคู่ส่วนโค้งแต่ละคู่ของอักขระที่ต้องการรู้จำกับอักขระต้นแบบ และ ค่าความแตกต่างระหว่างอักขระทั้งสอง ($DC_{II.MM}$) ด้วย

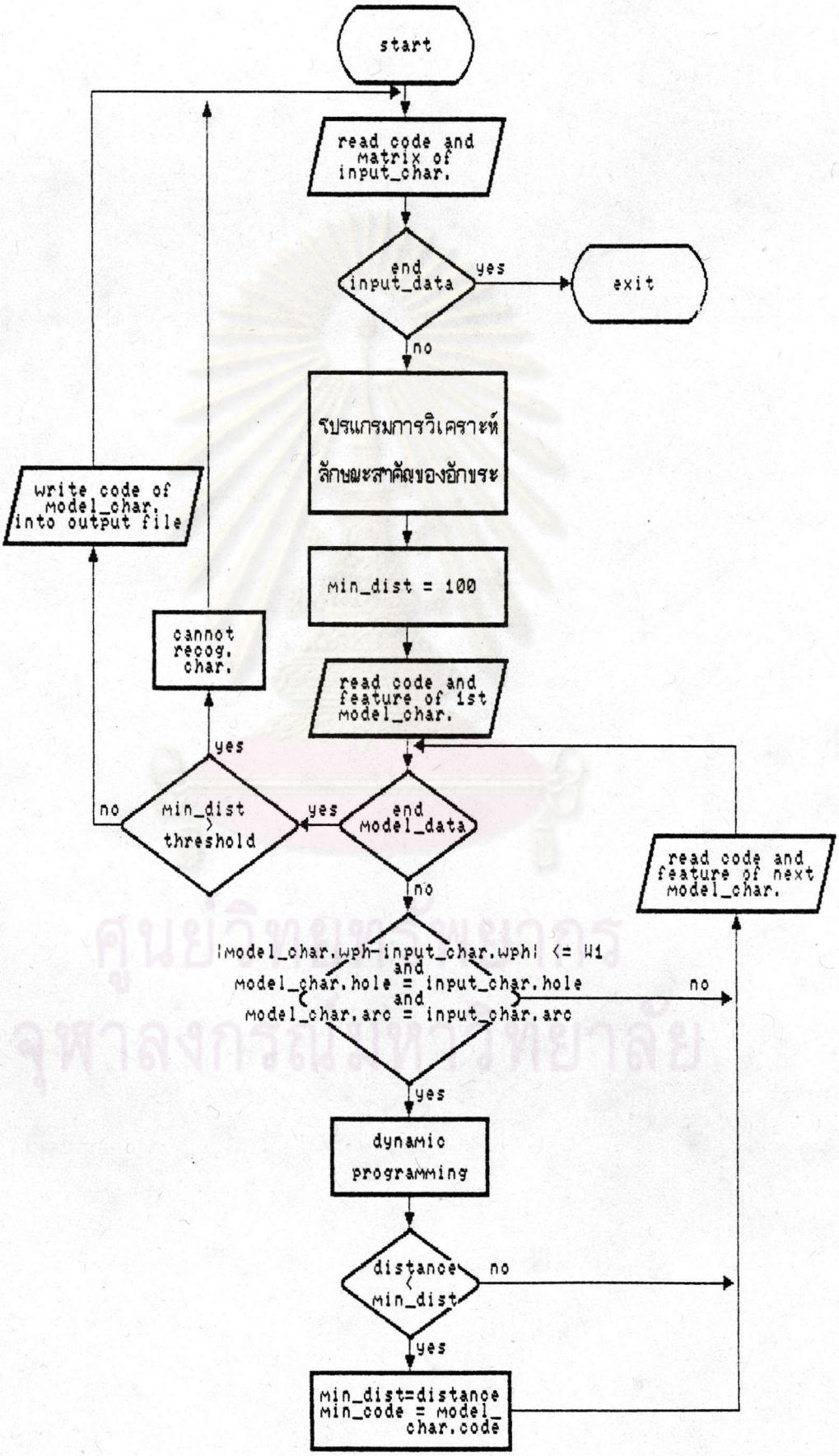
การหาค่าความแตกต่างระหว่างอักขระที่ต้องการรู้จำ กับอักขระต้นแบบนั้นจะต้องกระทำกับอักขระต้นแบบทุกตัวที่อยู่ในกลุ่มอักขระที่มีลักษณะคล้ายคลึงกัน เพื่อหาว่าอักขระต้นแบบตัวใดให้ค่าความแตกต่างที่น้อยที่สุดก็จะถือว่าอักขระต้นแบบตัวนั้นเป็นอักขระที่รู้จำได้ โดยค่าความแตกต่างที่น้อยที่สุดนี้จะต้องมีค่าไม่มากกว่าค่าที่กำหนดไว้ค่าหนึ่งด้วย มิฉะนั้นก็จะถือว่าอักขระที่ต้องการรู้จำนั้น เป็นอักขระที่ไม่สามารถบอกได้ว่าเป็นอักขระใด (rejected) นั่นก็คืออักขระที่มีค่า $DC_{II.MM}$ สอดคล้องกับเงื่อนไขต่อไปนี้ จะถือว่าเป็นอักขระที่รู้จำได้

(ก) $DC_{II.MM} < D_c$ เมื่อ D_c เป็นค่าที่กำหนดไว้ค่าหนึ่ง

(ข) $\min_{MM=1} (DC_{II.MM})$ เมื่อ MM = 1, ... , KM

โดย KM เป็นจำนวนอักขระต้นแบบทั้งหมดที่อยู่ในกลุ่มเดียวกันกับอักขระที่ต้องการรู้จำ

รูปที่ 3.37 ผังงานแสดงการหาอักขระต้นแบบที่มีความคล้ายกับอักขระที่ต้องการรู้จำมากที่สุด



ตารางที่ 3.3 ค่าความแตกต่างที่ได้ระหว่างคู่ส่วนโค้งของอักขระที่ต้องการรู้จำกับอักขระต้นแบบ และค่าความแตกต่างที่ได้ระหว่างอักขระทั้งสอง ($DC_{II,MM}$)

D(AL,AM)	ค่าที่ได้
D(A1,A1)	1.222
D(A2,A2)	1.250
D(A3,A3)	1.833
D(A4,A4)	1.179

ดังนั้นจะได้ว่า $DC_{II,MM}$ = (1.222 + 1.250 + 1.833 + 1.179) / 4

= 1.371

ในกรณีที่อักขระที่ต้องการรู้จำและอักขระต้นแบบมีจำนวนส่วนโค้งต่างกันไม่เกินกว่าค่าที่กำหนดไว้ค่าหนึ่ง ซึ่งอาจเป็นไปได้เนื่องจากอักขระตัวเดิมเมื่อนำไปกวาดตรวจด้วยแสงอีกครั้งหนึ่งแล้ว จะได้ข้อมูลที่แตกต่างจากเดิมดังที่ได้กล่าวถึงสาเหตุไว้แล้วในตอนต้น หรือในกรณีที่ค่าความแตกต่างที่น้อยที่สุดระหว่างอักขระที่ต้องการรู้จำกับอักขระต้นแบบมีค่ามากกว่าค่าที่กำหนดไว้ค่าหนึ่งแล้ว ก็จะนำอักขระที่ต้องการรู้จำนั้นมาเปรียบเทียบแบบไดนามิกโปรแกรมมิงอีกครั้งหนึ่ง ซึ่งการเพิ่มขั้นตอนของการเปรียบเทียบแบบไดนามิกโปรแกรมมิงในขั้นนี้เข้าไว้ด้วย จะมีส่วนช่วยทำให้ผลของการรู้จำมีความถูกต้องมากยิ่งขึ้น

ขั้นตอนของการเปรียบเทียบแบบไดนามิกโปรแกรมมิงในขั้นนี้ จะทำในลักษณะต่างจากเดิมเล็กน้อย โดยในขั้นนี้จะไม่มีการหาค่าความแตกต่างของส่วนโค้งทีละส่วนโค้ง แต่จะทำโดยการหาส่วนโค้งคู่ที่คล้ายกันมากที่สุดมาให้ได้ก่อน จากนั้นก็จะใช้คู่ส่วนโค้งนี้เป็นส่วนโค้งเริ่มต้น สำหรับหาค่าความแตกต่างระหว่างอักขระที่ต้องการรู้จำกับอักขระต้นแบบ

ดังนั้นสามารถจะแทนอักขระที่ต้องการรู้จำ กับ อักขระต้นแบบได้โดยใช้ลำดับของเวกเตอร์ที่แสดงลักษณะสำคัญ ดังต่อไปนี้

สมมุติให้ II แทน รูปแบบของอักขระที่ต้องการรู้จำ
และ MM แทน รูปแบบของอักขระต้นแบบ

ดังนั้นสามารถเขียนรูปแบบของอักขระในลักษณะของเวกเตอร์ได้ดังนี้

$$II = ii_1, ii_2, \ldots, ii_s, \ldots, ii_S$$
$$MM = mm_1, mm_2, \ldots, mm_t, \ldots, mm_T$$

โดยที่ ii_1 และ mm_1 เป็นฟีเชอร์แรกของส่วนโค้งคู่ที่คล้ายกันมากที่สุด

S และ T เป็นผลรวมของจำนวนจุดบ่งความนูนและจุดบ่งความเว้าของอักขระที่ต้องการรู้จำและอักขระต้นแบบ ตามลำดับ

และเช่นเดียวกับในขั้นตอนแรกของการเปรียบเทียบ คือ ii_s และ mm_t จะใช้แทนความยาวระหว่างจุดบ่งความนูนหรือจุดบ่งความเว้า 2 จุด ที่อยู่ติดกันบนส่วนโค้ง โดยที่ ii_1 ก็คือ ความยาวระหว่างจุดบ่งความนูนหรือจุดบ่งความเว้า 2 จุดแรกที่อยู่ติดกันบนส่วนโค้ง L คู่ที่คล้ายกันที่สุด และ mm_1 ก็คือ ความยาวระหว่างจุดบ่งความนูนหรือจุดบ่งความเว้า 2 จุดแรก ที่อยู่ติดกันบนส่วนโค้ง M คู่ที่คล้ายกันที่สุด ดังในรูปที่ 3.38

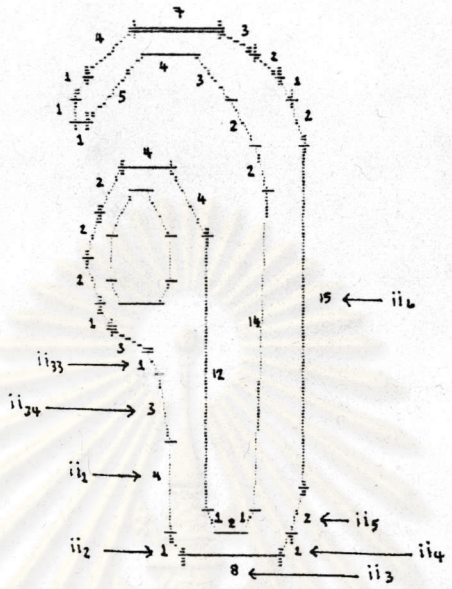
ดังนั้นค่าความแตกต่างระหว่างอักขระที่ต้องการรู้จำ กับ อักขระต้นแบบเป็นดังนี้

$$D(II,MM) = g(S,T) / NC$$

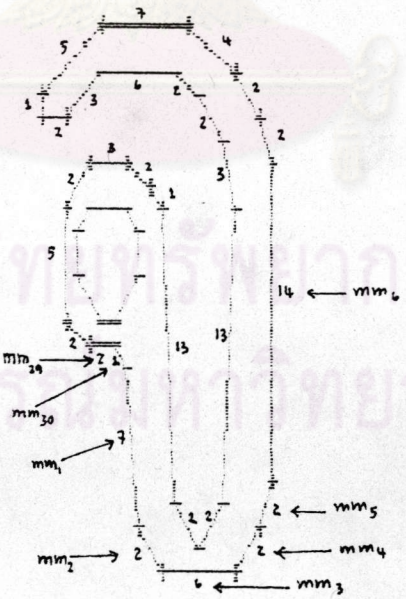
เมื่อ $NC = S + T$ ในกรณีของแกนเวลาสมมาตร
หรือ $NC = S$ ในกรณีของแกนเวลาไม่สมมาตร

อักขระต้นแบบ MM ซึ่งมีค่าความแตกต่าง D(II,MM) ที่น้อยที่สุดจะถือว่าเป็นอักขระต้นแบบที่รู้จำได้ ก็ต่อเมื่อค่าความแตกต่างที่น้อยที่สุดนี้มีค่าไม่เกินค่าที่กำหนดไว้ค่าหนึ่ง แต่ถ้าหากว่าค่าความแตกต่างที่น้อยที่สุดนี้มีค่ามากเกินกว่าค่าที่กำหนดไว้ค่าหนึ่งก็จะถือว่าอักขระที่ต้องการรู้จำนั้นเป็นอักขระที่ไม่สามารถบอกได้ว่าเป็นอักขระใด

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



(ก) อักขระที่ต้องการรู้จำ



(ข) อักขระต้นแบบ

รูปที่ 3.38 (ก) การกำหนดค่า ii_s ให้กับอักขระที่ต้องการรู้จำ
(ข) การกำหนดค่า mm_t ให้กับอักขระต้นแบบ