การใช้ AI ควบคุม AGV

AGV CONTROL USING AI

โดย

นายชัชวาล วัชรประภาพงศ์

นายสมชาย อัจฉริยวิทย์กุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

009614

ปริญญานิพนธ์ปีการศึกษา 2534

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เรื่อง การใช้ AI ควบคุม AGV

ผู้จัดทำ

1. นายชัชวาล วัชรประภาพงศ์ 31.1057
2. นายสมชาย อัจฉริยวิทย์กุล 31.1299

..... อาจารย์ที่ปรึกษา

(อ. ภากร หุตะสิงกาศ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	ก
Abstract	ข
บทที่ 1 บทนำ	1
บทที่ 2 เรื่องทั่วไปเกี่ยวกับปัญญาประดิษฐ์	2
บทที่ 3 เทคนิคของปัญญาประดิษฐ์	6
บทที่ 4 การแสดงความรู้ในปัญญาประดิษฐ์	27
บทที่ 5 โปรแกรมการควบคุม AGV	34
บทสรุปและวิจารณ์	
ภาคผนวก	
บรรณานุกรม	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ AI ควบคุม AGV

โดย นายชัชวาล วัชรประภาพงศ์ 31.1057

นายสมชาย อัจฉริยวิทย์กุล 31.1299

อาจารย์ที่ปรึกษา

อ. ภากร หุตะสังกาศ

บทคัดย่อ

การใช้ AI ควบคุม AGV นี้ เป็นการจำลองผังโรงงาน ทางเดินของ AGV และวัตถุกีดขวางบนจอคอมพิวเตอร์ โดยสามารถกำหนดเส้นทางเดินของ AGVและตำแหน่งของวัตถุกีดขวางได้ การแสดงความรู้ของ AGVจะใช้หลักการแสดงความรู้ด้วยกฎ(rule-base representation) และอนุมานโดยใช้หลักการอนุมานแบบเดินหน้า (Forward chaining) การเคลื่อนที่ของ AGV จะสามารถหลบสิ่งกีดขวางได้ โดยการตรวจจับของเซ็นเซอร์ ทางด้านหน้าและด้านข้างของตัวรถ

การใช้ AI ควบคุม AGV มีประโยชน์มากในการขนส่งภายในโรงงาน เนื่องจากสามารถหลบหลีกสิ่งกีดขวางและกำหนดเส้นทางได้ง่ายด้วยการโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AGV CONTROL USING AI

BY CHATCHAWAN WATCHARAPRAPAPONG 31.1057

SOMCHAI AJCHARIYAWITKUL 31.1299

ADVISOR

PHAKORN HUTASUNGKAS

ABSTRACT

AI-CONTROLLED AGV IS SIMULATED FOR AGV MOVING PATHS IN INDUSTRIAL PLANT AND AVOIDING OBSTRUCTION. THIS PROGRAM CAN BE SPECIFED THE COORDINATE POINT OF PATHS AND OBSTRUCTION. ALL KNOWLEDGE BASE CAN BE DEMONSTRATED BY RULE-BASE REPRESENTATION AND FORCASTING BY FORWARD CHAINING RULES. AGV CAN AVOID OBSTRUCTION BY SENSORS CONTACTED AT FRONT AND SIDE OF IT'S BODY

AGV CONTROL USING AI IS USED FOR TRANSPORTATION IN FACTORIES BECAUSE IT CAN AVOID OBSTRUCTION AND EASY TO PROGRAM FOR MOVEMENT.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

เนื่องจากเทคโนโลยีที่เจริญไปข้างหน้าอย่างรวดเร็ว ทำให้มีการพัฒนาเครื่องจักรต่างๆขึ้นมาทำงานแทนมนุษย์ AGV (Automated Guided Vehicle) ก็เป็นหนึ่งในหุ่นยนต์ที่มนุษย์สร้างขึ้นเพื่อใช้ในการเคลื่อนย้ายสิ่งของจากที่หนึ่งไปยังอีกที่หนึ่ง AGV เป็น Mobile Robot ที่สามารถเคลื่อนที่โดยการโปรแกรมทิศทาง เวลาและความเร็วในการเคลื่อนที่ ให้สามารถเคลื่อนที่ไปยัง node หรือ path ที่ต้องการ

AGV ถูกนำมาใช้ในการเคลื่อนย้าย หรือบรรทุกสิ่งของในโรงงานอุตสาหกรรม เนื่องจากมีความคล่องตัวสูง และมีประสิทธิภาพสูงกว่าการใช้ conveyor มาก นอกจากในงานอุตสาหกรรมแล้ว AGV ยังถูกนำไปใช้ในพื้นที่ที่เสี่ยงอันตรายสำหรับมนุษย์อีกด้วย

สำหรับในส่วนของโครงงานที่ทำนั้น จะต้องทำให้ AGV สามารถหลบหลีกสิ่งกีดขวางที่อยู่บนทางเดิน โดยใช้การเคลื่อนที่แบบเวลาจริง (real-time) โดยจะใช้ AI เชื่อมโยงเข้ากับส่วนของ AGV ในการเคลื่อนที่นั้น มีค่าตัวแปรที่จะต้องนำมาพิจารณาอยู่ 4 ตัว คือ

1. ความกว้างของทาง
2. ขนาดของสิ่งกีดขวาง
3. ตำแหน่งของสิ่งกีดขวาง
4. ความเร็วของ AGV ขณะวิ่งมา

อธิบายได้ว่าขณะที่ AGV วิ่งมานั้น จะต้องมีการตรวจสอบค่าตัวแปรทั้ง 4 ข้างต้น เพื่อนำมาประมวลผลในการเคลื่อนที่เข้าหาจุดหมายที่กำหนด หากพบสิ่งกีดขวางจะต้องทำการหลบหลีกให้พ้น โดยใช้ AI ช่วยในการตัดสินใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

เรื่องทั่วไปเกี่ยวกับปัญญาประดิษฐ์

บทนำ

ปัญญาประดิษฐ์ (Artificial Intelligence : AI) เป็นสาขาหนึ่งของศาสตร์ทางคอมพิวเตอร์ ที่เกี่ยวข้องกับการทำให้คอมพิวเตอร์สามารถทำงานได้คล้ายมนุษย์ ที่สามารถประมวลผลในลักษณะของการหาเหตุผล การตัดสินใจ การแก้ปัญหา หรือการทำให้คอมพิวเตอร์สามารถแสดงความฉลาดออกมาได้ เช่น การคิด การหาเหตุผล การรับรู้ หรือการกระทำ ในการที่จะทำให้คอมพิวเตอร์สามารถทำงานเช่นนี้ได้นั้น จะต้องพัฒนาคอมพิวเตอร์ให้มีความเร็วในการประมวลผลสูง และจะต้องมีโปรแกรมที่สามารถทำการประมวลผลอย่างมีประสิทธิภาพ

สถาบันที่ทำงานวิจัยทางด้านปัญญาประดิษฐ์ในระยะแรกได้แก่Massachusetts Institute of Technology มักจะทำการวิจัยเรื่องที่เกี่ยวข้องกับหุ่นยนต์ (robotics) การมองเห็น (vision sennsing) การหาเหตุผล (reasoning) การเขียนโปรแกรม (programming) และการออกแบบระบบผู้เชี่ยวชาญ (expert system design) เป็นต้น

สิ่งที่เรียกได้ว่าเป็นความสำเร็จทางปัญญาประดิษฐ์ คือการพัฒนาระบบผู้เชี่ยวชาญที่เกิดขึ้นเมื่อประมาณกลางทศวรรษ 1960 เมื่อนักวิจัยทางปัญญาประดิษฐ์หันมาสนใจการสร้างโปรแกรมที่มีประสิทธิภาพสูง และการแก้ปัญหาพิเศษเฉพาะด้าน โดยอาศัยความรู้ที่จะต้องรวบรวมจากผู้เชี่ยวชาญที่เป็นมนุษย์

เทคนิคของปัญญาประดิษฐ์

ในปี 2493 ได้มีการเสนอวิธีการในการทดสอบสิ่งที่เรียกว่าปัญญาประดิษฐ์ขึ้น มีชื่อเรียกว่าการทดสอบของทิวริง (Turing test) วิธีการนี้จะต้องอาศัยคน 2 คน และเครื่องมือที่จะทดสอบโดยที่คนหนึ่งจะทำหน้าที่เป็นผู้ซักถามและผู้ซักถามนี้จะถูกแยกออกจากคนอีกคนหนึ่งและเครื่องมือที่จะทดสอบ ผู้ซักถามจะต้องตั้งคำถามถามเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือคนก็ได้โดยที่คนถามไม่ทราบเลยว่าอะไรคือเครื่องหรือคน อาจจะทราบเพียงแต่ว่าเป็น A หรือ Bเท่านั้น หน้าที่ของผู้ซักถามจะต้องถามทั้ง A และ B เพื่อจำแนกให้ได้ว่า A และ B คือเครื่องหรือคน ถ้าเครื่องที่นำเข้ามาทำการทดสอบสามารถหลอกผู้ซักถามได้ว่าเป็นคน เครื่องนั้นก็อาจพิจารณาได้ว่ามีความสามารถในการคิดหาเหตุผล

ในการหลอกของเครื่อง เครื่องจะต้องทำหน้าที่ให้เหมือนคนมากที่สุดเช่น ถ้าถูกออกคำสั่งให้คำนวณเลข (84572+45812)-2546870 เครื่องจะให้คำตอบโดยทันทีไม่ได้ เพราะพฤติกรรมของคนมิได้เป็นเช่นนั้น

ถ้าพิจารณาถึงระบบที่เป็นปัญญาประดิษฐ์จริง เช่นเดนดรอลซึ่งเป็นระบบที่ใช้ในการวิเคราะห์โครงสร้างของสารประกอบชีวภาพ เมื่อมองจากระบบนี้แล้วจะเห็นว่ายากที่จะหาเครื่องวัดอะไรมาทำการทดสอบหาระดับของความใกล้เคียงระหว่างระบบและนักเคมี แต่อย่างไรก็ตาม ผลของการวิเคราะห์ที่เกิดจากเดนดรอล ก็ได้รับการตีพิมพ์และยอมรับจากนักเคมีทั่วไป ซึ่งลักษณะที่ว่านี้คือ AI ชนิดหนึ่งนั่นเอง

งานวิจัยในอดีตที่ผ่านมากว่า 20 ปี ทางด้านปัญญาประดิษฐ์ได้ให้ข้อสรุปไว้ว่า "ความฉลาดนั้นได้มาจากความรู้" ดังนั้นเทคนิคเบื้องต้นในการทำให้เครื่องมือมีความฉลาดขึ้นมาได้ก็คือเทคนิคในการใช้ประโยชน์จากความรู้นั้นเอง ในการกำหนดรูปแบบของความรู้นั้นจะต้องมีลักษณะต่างๆ ดังต่อไปนี้

1. ความรู้นั้นจะต้องมีลักษณะทั่วไป หรืออีกความหมายหนึ่งก็คือความครอบคลุมไปถึงความรู้อื่นๆที่มีลักษณะคล้ายคลึงกันได้ด้วย ถ้ามิฉะนั้นแล้ว จะต้องมีหน่วยความจำจำนวนมหาศาลเพื่อมาเก็บความรู้

2. ความรู้นั้นจะต้องอยู่ในรูปแบบที่ผู้ให้ความรู้กับระบบ สามารถที่จะเข้าใจได้และง่ายต่อการทำงานของเครื่องด้วย

3. ความรู้นั้นจะต้องง่ายต่อการแก้ไขปรับปรุงหรือเพิ่มเติมได้เมื่อต้องการ

4. ความรู้นั้นจะต้องนำไปใช้ได้กับหลายๆสถานการณ์

5. ความรู้นั้นจะต้องอยู่ในรูปแบบที่สามารถนำไปใช้ประโยชน์ในการแก้ปัญหาเรื่องใดเรื่องหนึ่งได้ มิใช่ความรู้ใดๆที่กระจัดกระจาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แม้ว่าเทคนิคดังกล่าวมาแล้วเป็นเรื่องที่สำคัญและจำเป็น แต่ในทางความเป็นจริง เทคนิคเหล่านี้ต้องมีขอบเขตของการใช้งาน ซึ่งอาจจะเป็นไปได้ว่า ปัญหาของปัญญาประดิษฐ์บางปัญหาไม่จำเป็นที่จะต้องใช้เทคนิคนี้ และเช่นกัน เทคนิคเหล่านี้ก็สามารถที่จะนำไปแก้ปัญหาบางอย่างที่ไม่ใช่เรื่องของปัญญาประดิษฐ์ได้

การประยุกต์ใช้งานปัญญาประดิษฐ์

ในเรื่องของปัญญาประดิษฐ์ การประยุกต์ใช้งานเฉพาะด้าน ยังเป็นเรื่องหลักของการพัฒนา ซึ่งโดยปกติแล้วจะจำแนกออกเป็นได้หลายด้านมากมาย โดยเฉพาะทางด้านอุตสาหกรรม เช่น การพัฒนาคอมพิวเตอร์ช่วยออกแบบ คอมพิวเตอร์ช่วยสอน คอมพิวเตอร์ควบคุมระบบอุตสาหกรรมอัตโนมัติ การทำความเข้าใจภาพ และระบบช่วยเขียนโปรแกรมคอมพิวเตอร์ เป็นต้น

สำหรับเรื่องที่มีการประยุกต์ใช้งานมากที่สำคัญ 2 เรื่องได้แก่

1. **การประมวลผลภาษาธรรมชาติ** (natural language processing) เป็นการประยุกต์ใช้ปัญญาประดิษฐ์ที่สำคัญมากสาขาหนึ่ง ในสาขานี้ปัญญาประดิษฐ์ได้ถูกนำมาใช้เพื่อสามารถทำความเข้าใจภาษามนุษย์โดยคอมพิวเตอร์ พื้นฐานของการวิจัยนี้เป็นเรื่องที่เกี่ยวกับการประยุกต์ใช้ภาษาศาสตร์และคอมพิวเตอร์เข้าด้วยกัน โดยจะหาวิธีที่ให้คอมพิวเตอร์สามารถเข้าใจในเรื่องของคำและความหมายของคำไวยากรณ์ของประโยคและความหมายของทั้งประโยค ผลของการวิจัยจะออกมาในหลายรูปแบบ เช่นการใช้คำสั่งให้เครื่องจักรทำงานโดยใช้ภาษาอังกฤษ การแปลภาษาด้วยเครื่องคอมพิวเตอร์ เป็นต้น

2. **ระบบผู้เชี่ยวชาญ** (expert systems) เป็นระบบที่ถูกออกแบบขึ้นมาเพื่อทำหน้าที่ในการเป็นผู้เชี่ยวชาญและให้คำปรึกษากับมนุษย์ในเรื่องต่างๆ

ในการพัฒนาระบบปัญญาประดิษฐ์ มีปัญหาพื้นฐานที่ผู้พัฒนาจะต้องทำความเข้าใจ คือ มีเทคนิคในการสร้างโปรแกรมอย่างไรที่จะทำให้คอมพิวเตอร์สามารถประมวลผลดังกล่าวได้ คำตอบก็คือ ระบบปัญญาประดิษฐ์นั้นจะต้องมีความรู้อยู่ด้วย เมื่อนำมาประกอบกับการประมวลผลด้วยคอมพิวเตอร์แล้วระบบนี้ก็จะสามารถแก้ปัญหาบางอย่างได้เช่นเดียวกับมนุษย์แต่ในการทำให้คอมพิวเตอร์มีความรู้และสามารถทำการประมวล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลความรู้ได้ไม่ใช่เรื่องง่าย ทั้งนี้เพราะความรู้เป็นฐานข้อมูลที่มีขนาดใหญ่มาก และการแก้ปัญหาของมนุษย์ก็เป็นกระบวนการที่ซับซ้อน

ข้อสังเกตในการแก้ปัญหา

ระบบการผลิต เป็นกระบวนการในการวางโครงสร้างโปรแกรมแบบ AI ให้มีลักษณะง่ายต่อการอธิบายกระบวนการ(หรือวิธีการแก้ปัญหา)ซึ่งจะต้องประกอบไปด้วย

1. กฎ ซึ่งกฎแต่ละข้อต้องประกอบด้วยส่วนที่อยู่ทางซ้ายที่จะอธิบายถึงเงื่อนไข และส่วนที่อยู่ทางขวาที่อธิบายถึงผลของกฎ

2. ฐานข้อมูลที่มีข้อมูลที่ต้องการ บางส่วนของฐานข้อมูลจะเป็นแบบถาวร และบางส่วนจะเป็นสิ่งที่เกี่ยวข้องกับการแก้ปัญหาในช่วงนั้น

3. กลไกในการควบคุม เป็นส่วนที่กำหนดลำดับของกฎที่จะนำมาใช้หรือเปรียบเทียบกับฐานข้อมูลเพื่อบอกลำดับของการแก้ปัญหา สำหรับกลไกในการควบคุมนี้มีอยู่ 3 สิ่งที่สำคัญ คือ

- การกำหนดทิศทางสำหรับการค้นหา
- กระบวนการในการเลือกกฎ
- การค้นหาแบบฮิวริสติก

ในการแก้ปัญหาด้วยปัญญาประดิษฐ์ การแก้ปัญหาจะต้องประกอบด้วยส่วนที่สำคัญ 2 ส่วน คือส่วนที่ทำหน้าที่เกี่ยวกับการอนุมาน หรือการกระบวนการของการค้นหาคำตอบดังที่ได้กล่าวมาแล้ว และส่วนที่เป็นเรื่องเกี่ยวกับความรู้หรือกฎฮิวริสติกเป็นส่วนที่กระบวนในการค้นหาคำตอบจะต้องนำไปใช้ในการอนุมาน ส่วนนี้จะประกอบด้วยกฎความจริง และข้ออ้างอิงต่างๆที่อธิบายเกี่ยวกับเรื่องที่ต้องการแก้ปัญหาซึ่งผู้เชี่ยวชาญหรือบุคคลที่มีความรู้จะต้องเป็นผู้เขียนขึ้นมา การเขียนความรู้นี้จะเรียกว่าการแสดงความรู้ (knowledge representation) สำหรับการแสดงความรู้นี้จะต้องมีหลักการเช่นเดียวกับการเขียนโปรแกรมคอมพิวเตอร์ คือผู้เชี่ยวชาญจะต้องยึดหลักโครงสร้างในการเขียน โครงสร้างนี้จะต้องง่ายกับผู้เขียน และคอมพิวเตอร์จะต้องเข้าใจได้สำหรับความรู้จะเป็นการอธิบายถึงเรื่องกฎ ความจริงและข้ออ้างต่างๆ แต่การเขียนโปรแกรมจะเป็นการอธิบายถึงกระบวนการในการแก้ปัญหาตามที่ผู้เขียนต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

เทคนิคของปัญญาประดิษฐ์

การกำหนดนิยามให้กับปัญหา

เพื่อเป็นการทำความเข้าใจในเทคนิคของปัญญาประดิษฐ์เบื้องต้นในที่นี้จะแสดงให้เห็นถึงวิธีในการแก้ปัญหา และหลักการบางอย่างในการกำหนดหลักเกณฑ์ของปัญหา วิธีการและหลักการดังกล่าว เป็นกระบวนการพื้นฐานที่จะทำให้เข้าใจการแก้ปัญหาในเชิงคอมพิวเตอร์

กำหนดให้มีคนโท 2 ใบที่มีขนาดใส่น้ำ 4 และ 3 แกลลอนตามลำดับ จะมีวิธีการอย่างไรที่จะทำให้คนโทที่มีขนาด 4 แกลลอน มีน้ำอยู่ 2 แกลลอนพอดี โดยที่ไม่ต้องอาศัยเครื่องวัดใดๆทั้งสิ้น

จากโจทย์ดังกล่าวข้างต้น การแก้ปัญหามีด้วยกันหลายวิธี แต่ในที่นี้จะอาศัยวิธีการของห้วงสถานะมาช่วยในการแก้ โดยที่กำหนดจุดเริ่มต้นของปัญหาและเป้าหมายของการแก้ปัญหาว่าคืออะไร สำหรับเรื่องคนโทน้ำ การแก้ปัญหาจะเริ่มต้นจาก คนโทน้ำ 2 ใบที่ไม่มีน้ำเลย และผลลัพธ์สุดท้ายจะต้องเป็น มีน้ำอยู่ 2 แกลลอนในใบแรก และใบที่ 2 มีน้ำเท่าไรก็ได้

ถ้ากำหนดว่า X ซึ่งเป็นตัวแปรใดๆที่แทนค่าของปริมาณน้ำในคนโทในใบแรก และ Y คือตัวแปรใดๆที่แทนค่าของปริมาณน้ำในคนโทใบที่สอง

ดังนั้นค่าต่างๆที่เป็นไปได้ทั้งหมดของ X และ Y จะเป็นดังนี้

X = 0,1,2,3,4

Y = 0,1,2,3

และปริมาณน้ำในคนโท 2 ใบ สามารถเขียนอยู่ในรูปของ X และ Y ได้เป็น (X,Y) จุดเริ่มต้นในการแก้ปัญหาเริ่มจาก (0,0) ซึ่งหมายถึงสถานะที่ไม่มีน้ำในคนโท สำหรับเป้าหมายที่ต้องการคือ (2,n) โดย n เป็นเลขใดๆที่น้อยกว่า 3

ในการแก้ปัญหา จะต้องทำการเปลี่ยนสถานะจาก (0,0) ไปเป็น (2,n) ซึ่งจะทำได้โดยการอาศัยกฎเกณฑ์ต่างๆที่เกี่ยวกับการถ่ายเทน้ำของปัญหานี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กฎเกณฑ์เหล่านี้ได้แก่

1. เติมน้ำลงในคนโทใบแรกจนเต็ม
2. เติมน้ำลงในคนโทใบสองจนเต็ม
3. เทน้ำบางส่วนออกจากใบแรก
4. เทน้ำบางส่วนออกจากใบที่สอง

ถ้าหากอธิบายกฎเหล่านี้ในรูปของตัวแปร X และ Y จะได้ดังรูป 2.1

กฎ	คำอธิบาย
1. (X,Y : X < 4) -> (4,Y)	เติมน้ำลงในคนโทใบแรกจนเต็ม
2. (X,Y : Y < 3) -> (X,3)	เติมน้ำลงในคนโทใบสองจนเต็ม
3. (X,Y : X > 0) -> (X-D,Y)	เทน้ำบางส่วนออกจากใบแรก
4. (X,Y : Y > 0) -> (X,Y-D)	เทน้ำบางส่วนออกจากใบสอง
5. (X,Y : X > 0) -> (0,Y)	เทน้ำออกจากใบแรกจนหมด
6. (X,Y : Y > 0) -> (X,0)	เทน้ำออกจากใบสองจนหมด
7. (X,Y : X+Y >= 4^Y > 0) -> (4,Y-(4-X))	นำน้ำจากคนโทใบที่ 2 ใส่ใบแรกจนเต็ม โดยที่ คนโททั้ง 2 ใบมีน้ำรวมกันมากกว่า 4 แกลลอน
8. (X,Y : X+Y >= 3^X > 0) -> (X-(3-Y),3)	เทน้ำจากคนโทใบแรกใส่ใบที่ 2 จนเต็ม โดยที่ คนโท 2 ใบมีน้ำรวมกันมากกว่า 3 แกลลอน
9. (X,Y : X+Y <= 4^Y > 0) -> (X+Y,0)	เทน้ำจากคนโทใบที่ 2 มาใส่ในใบที่ 1
10. (X,Y : X+Y <= 3^X > 0) -> (0,X+Y)	เทน้ำจากคนโทใบที่ 1 ใส่ใบที่ 2

รูปที่ 2.1 กฎที่อธิบายการแก้ปัญหาเรื่องคนโทน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อสังเกตในการกำหนดนิยามของปัญหา

1. ในการเขียนกฎเพื่ออธิบายปัญหาหนึ่งจะต้องมีองค์ประกอบพื้นฐานอย่างน้อย 2 ส่วนคือ ส่วนที่ใช้ตรวจสอบเงื่อนไข (condition) และข้อสรุป (conclusion) ส่วนที่เป็นเงื่อนไข คือส่วนที่ใช้ตรวจสอบเพื่อนำกฎข้อนั้นมาใช้งาน และส่วนที่เป็นข้อสรุปคือส่วนที่ใช้งานของกฎ เช่น

(X,Y/X<4)-> (4,Y)

หมายความว่าภายใต้เงื่อนไขของ X และ Y ที่มี X น้อยกว่า 4 แกลลอน (X,Y /X<4) จะสรุปว่ามีน้ำอยู่ 4 แกลลอนในคนโทแรก และมีน้ำไม่ทราบจำนวนในคนโทใบที่สอง (4,Y)

ลักษณะของกฎที่แสดงข้างต้นเรียกว่า กฎการผลิต (production rule) ซึ่งกฎการผลิตนี้จะต้องกำหนดหรือสร้างกฎจำนวนหนึ่งที่สามารถอธิบายการกระทำทั้งหมดที่เป็นไปได้ของปัญหานั้นๆ

2. การกำหนดค่าของ X และ Y จะต้องสามารถควบคุมถึงความเป็นไปได้ทั้งหมด ซึ่งค่าของ X และ Y นี้จะเรียกว่า สถานะ (state)

ในที่นี้เรากำหนดสถานะต่างๆ เป็น

X =0,1,2,3,4 และ Y =0,1,2 หรือ 3

3. ในการกำหนดขั้นตอนต่างๆของการแก้ปัญหาเรียกว่าห้วยสถานะ ซึ่งกำหนดว่าห้วงหนึ่งๆอย่างน้อยจะต้องมี 2 สถานะ คือ

สถานะเริ่มต้น เป็น (0,0) และ

สถานะเป้าหมาย เป็น (2,n)

การแก้ปัญหาสำหรับคนโทน้ำ

การแก้ปัญหาเป็นการนำกฎตามที่กล่าว มาอธิบายการเปลี่ยนแปลงต่างๆของสถานะ จากสถานะเริ่มต้นจนถึงสถานะเป้าหมาย ในการแก้ปัญหาตามลักษณะนี้ จะเรียกว่าระบบการผลิต (production system) จากปัญหาดังกล่าวข้างต้นสามารถเขียนตารางของการแก้ปัญหาได้ดังรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คนโท 1	คนโท 2	กอปปี้	สถานะ
0	0	-	สถานะเริ่มต้น
0	3	2	
3	0	9	
3	3	2	
4	2	7	
0	2	5	
2	0	9	สถานะเป้าหมาย

รูปที่ 2.2 แสดงการแก้ปัญหาเรื่องคนโทน้ำ

ข้อสังเกตในการแก้ปัญหา

ระบบการผลิต เป็นกระบวนการในการวางโครงสร้างโปนแกรมแบบ AI ให้มีลักษณะง่ายต่อการอธิบายกระบวนการ (หรือวิธีการแก้ปัญหา) ซึ่งจะต้องประกอบไปด้วย

1. กฎ ซึ่งกฎแต่ละข้อต้องประกอบด้วยส่วนที่อยู่ทางซ้ายที่จะอธิบายถึงเงื่อนไขของกฎ และส่วนที่อยู่ทางขวาที่อธิบายถึงผลของกฎ

2. ฐานข้อมูลที่มีข้อมูลที่ต้องการ บางส่วนของฐานข้อมูลจะเป็นแบบถาวร และบางส่วนจะเป็นสิ่งที่เกี่ยวข้องกับการแก้ปัญหาในช่วงนั้น

3. กลไกในการควบคุม เป็นส่วนที่กำหนดลำดับของกฎที่จะนำมาใช้หรือเปรียบเทียบกับฐานข้อมูล เพื่อบอกลำดับของการแก้ปัญหา สำหรับกลไกในการควบคุมนี้มีอยู่ 3 สิ่งที่สำคัญ คือ

- การกำหนดทิศทางสำหรับการค้นหา
- กระบวนการในการเลือกกฎ
- การค้นหาแบบฮิวริสติก

วิธีต่างๆในการแก้ปัญหา

จากตัวอย่างที่กล่าวมาถึงกระบวนการในการเลือกและจัดเรียงลำดับกฎนั้นเป็นกระบวนการหนึ่งของการแก้ปัญหา หรือถ้าจะมองอีกนัยหนึ่งก็คือการหาเหตุผล ในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

009614

แก้ปัญหาหรือการหาเหตุผลนี้ กระบวนการของคอมพิวเตอร์ที่จะต้องกระทำคือการค้นหาข้อมูล (search) ดังตัวอย่างเรื่องคนโทน้ำ 2 ใบนั้น เป็นเพียงตัวอย่างของการแสดงให้เห็นถึงวิธีการของการค้นหาข้อมูลที่ยังมีความรู้ไม่มากนักถ้าหากมองปัญหาที่ใหญ่ขึ้น ก็จะเป็นเรื่องยากมากที่จะอาศัยการค้นหาคำตอบ ด้วยวิธีของการค้นหาข้อมูลธรรมดา

ในการแก้ปัญหาของปัญญาประดิษฐ์นั้น มีเรื่องใหญ่ที่จะต้องพิจารณาถึง คือ

- การกำหนดทิศทางสำหรับการค้นหา
- รูปแบบของโครงสร้างข้อมูลที่ใช้สำหรับการค้นหา
- การแสดงความรู้
- กระบวนการในการเลือกกฎ
- การค้นหาแบบฮิวริสติก

การกำหนดทิศทางสำหรับการค้นหา

โดยปกติแล้วการกำหนดทิศทางของการค้นหามี 2 อย่าง คือ

- การค้นหาแบบเดินหน้า (forward reasoning)
- การค้นหาแบบย้อนหลัง (backward reasoning)

การหาเหตุผลแบบเดินหน้าและย้อนหลัง เป็นกระบวนการของการกำหนดทิศทางสำหรับการค้นหาและการเลือกกฎข้อที่เหมาะสม เพื่อที่กำหนดทิศทางว่าในการอนุมาน จะเริ่มต้นค้นหาคำตอบจากส่วนใดของห้วงปัญหา และการจัดเตรียมกฎหรือฐานความรู้ให้เหมาะกับการค้นหาคำตอบแบบนั้นโดยปกติแล้วจะอาศัยโครงสร้างข้อมูลแบบต้นไม้ (tree structure) และ\หรือแบบกราฟ (graph) เป็นรูปแบบหลัก

การหาเหตุผลแบบเดินหน้า จะเริ่มต้นจากสถานะเริ่มต้นนี้มาเป็นราก (root) ของโครงสร้างต้นไม้ และสร้างโหนดลูก (successor) ในระดับถัดไป โดยการหากฎทุกข้อ ซึ่งทางด้านซ้ายของกฎที่ตรงกับกับรูตโหนด และใช้ทางด้านขวาของกฎสร้างเป็นลูกของโหนดนั้น และสร้างระดับต่อไปด้วยการหยิบโหนดลูกนั้นมาทำด้วยวิธีเดียวกันจนครบทุกโหนดจนหมดในระดับนั้น และให้ทำเช่นนี้ไปเรื่อยๆจนครบกฎทุกข้อ

การหาเหตุผลแบบย้อนหลัง จะเริ่มต้นจากสถานะเป้าหมาย โดยการนำเอาเป้าหมายนั้น มาทำเป็นรากของโครงสร้างต้นไม้ และสร้างลูกในระดับถัดไป โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหากฎทุกข้อ ซึ่งทางด้านขวาของกฎตรงกับรูตโหนด และใช้ทางด้านซ้ายของกฎสร้างเป็นลูกของโหนดนั้น และสร้างระดับต่อไปด้วยการหยิบลูกโหนดนั้นมาทำด้วยวิธีเดียวกันจนครบทุกโหนด จนหมดในระดับนั้น และให้ทำเช่นนี้ไปเรื่อยๆจนครบกฎทุกข้อ

รูปแบบของโครงสร้างข้อมูลที่ใช้สำหรับการค้นหา

ในการกำหนดโครงสร้างข้อมูลสำหรับการค้นหานั้น โดยส่วนใหญ่แล้วจะถูกกำหนดให้อยู่ในรูปของโครงสร้างต้นไม้หรือโครงสร้างกราฟ ซึ่งโครงสร้างทั้งสองนี้สามารถเปลี่ยนแปลงไปมาได้ ตัวอย่างของโครงสร้างข้อมูลแบบต้นไม้และกราฟได้แสดงไว้ดังรูป 2.3 ซึ่งเป็นเรื่องของคนโทน้ำ

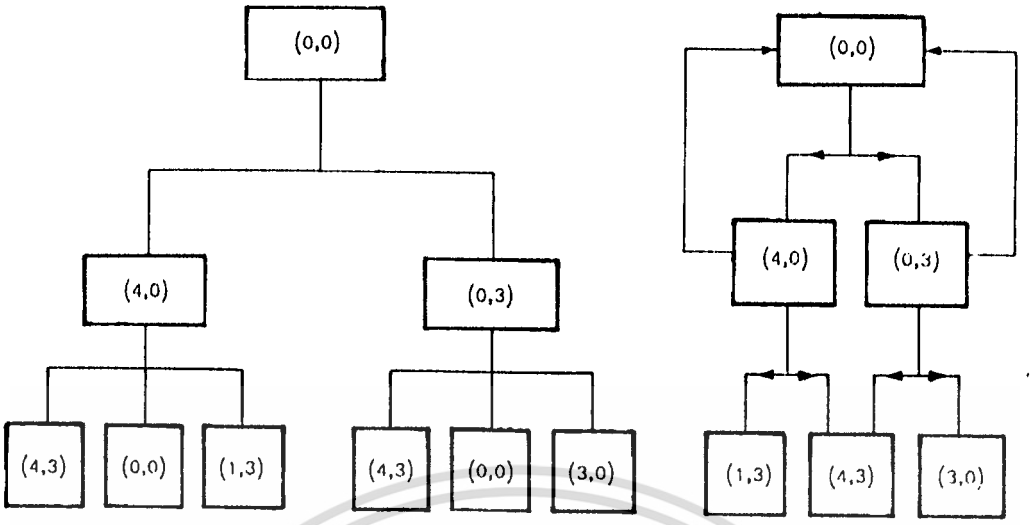
การสร้างรูปแบบโครงสร้างข้อมูลแบบกราฟ โดยทั่วไปแล้วจะดีกว่าแบบต้นไม้ เนื่องจากจะทำให้ประหยัดเนื้อที่ของหน่วยความจำและการค้นหาข้อมูลทำได้เร็ว แต่ข้อเสียก็คือ การเชื่อมต่อของข้อมูลทำได้ยากและใช้เวลาในการเชื่อมต่อข้อมูลนาน ดังนั้น ในการออกแบบโครงสร้างบางครั้งจึงนิยมสร้างเป็นแบบต้นไม้ก่อน แล้วจึงค่อยเปลี่ยนเป็นแบบกราฟ

การเปลี่ยนกระบวนการค้นหาในโครงสร้างต้นไม้(treesearch procedure) เป็นกระบวนการค้นหาในโครงสร้างกราฟ (graph search procedure)สามารถทำได้ด้วยวิธีการดังต่อไปนี้

1. ตรวจสอบเซตของโหนดที่ได้สร้างมาแล้วว่าโหนดที่จะสร้างใหม่มีอยู่แล้วหรือยัง
2. ถ้ายังไม่มี ก็สร้างโหนดใหม่ในระดับถัดไปให้กับโครงสร้างต้นไม้
3. ถ้ามีให้ทำดังต่อไปนี้

- ตั้งให้โหนดที่จะสร้างใหม่ชี้ไปยังโหนดที่มีอยู่แล้ว และเอาโหนดที่จะเกิดทิ้งไป
- ในการตรวจสอบเส้นทางที่สั้นที่สุด ให้ทำการตรวจสอบทุกระยะที่มีการสร้างเส้นทางขึ้นใหม่ แล้วเลือกเส้นทางที่สั้นที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) โครงสร้างข้อมูลแบบต้นไม้ (ข) โครงสร้างข้อมูลแบบกราฟ

รูปที่ 2.3 รูปของโครงสร้างข้อมูลแบบต้นไม้และกราฟ

การแสดงความรู้

เป็นกระบวนการในการแสดงความหมายที่ปรากฏอยู่ในแต่ละโหนดว่าจะมีวิธีการเช่นไร ในกรณีของคนโทน้ำดังที่ได้กล่าวนี้ อาศัยวิธีการแสดงความหมายโดยใช้ชุดของตัวเลข โดยตัวแรกแสดงระดับน้ำของคนโทใบแรก และตัวหลังแสดงระดับน้ำของคนโทใบที่สองดังนี้

วิธีการดังกล่าวเป็นการแสดงความรู้อย่างง่ายๆ แต่ในกรณีที่ความรู้มีความซับซ้อนมากขึ้น มีเรื่องหลายอย่างที่จะต้องพิจารณา เช่น ในกรณีที่ความรู้มิใช่ตัวเลขแต่เป็นออบเจ็คต์ (object) และความจริง (fact) ที่มีความสัมพันธ์กัน เช่น ความรู้ " plant is on the table " จะมีออบเจ็คต์ 2 ตัว คือ plant และ table

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่มี on แสดงถึงความสัมพันธ์ การแสดงความรู้แบบนี้ จะมีตัวอย่างดังต่อไปนี้

ON(plant,table) : plant is on the table

IN(table,room) : table is in the room

UNDER(table,window) : table is under the window

โดยปกติแล้วการแสดงความรู้ที่เป็นแบบออบเจ็คต์ที่เชื่อมต่อกันด้วยตัวแสดงความสัมพันธ์นี้เป็นวิธีการที่ง่าย แต่จะมีข้อที่จะต้องระวังคือ

1. ความรู้ทั้งหมดจะสามารถรวมเป็นความรู้เดียวกันได้อย่างไร เช่นหากว่าเรากำลังอธิบายถึงลักษณะของห้องห้องหนึ่ง ที่บอกว่า ห้องนี้ตั้งโต๊ะไว้ใต้หน้าต่าง " table is under the window " แล้ววันหนึ่งเมื่อมีการเปลี่ยนฐานความรู้ว่า CENTER(table,room) ในระบบการแสดงความรู้ จะมีวิธีการอย่างไรที่จะทำให้ทราบว่า UNDER(table,window) ใช้ไม่ได้แล้ว เพราะเมื่อโต๊ะมาอยู่กลางห้องก็เป็นไปไม่ได้ที่โต๊ะตัวเดียวกันจะอยู่ใต้หน้าต่างด้วย

2. การจัดลำดับอย่างไรทำให้การค้นหาทำได้ง่าย เช่น ถ้าเราจะเติมคำว่า ABOVE(ceiling,floor)เข้าไปในฐานความรู้ จะใส่ตรงไหนที่จะทำให้ไม่ต้องบอกทุกครั้งเมื่อมีการกล่าวอ้างถึงเรื่องของห้อง เพราะ"เพดานอยู่เหนือพื้น" นี้เป็นความจริงทั่วไปในเรื่องที่เกี่ยวกับห้อง

กระบวนการในการเลือกกฎ

โดยปกติแล้ววิธีในการเลือกกฎสามารถทำได้โดยวิธีการเปรียบเทียบ(maching) ซึ่งทำได้โดยการนำสถานะปัจจุบัน ไปเปรียบเทียบกับส่วนของเงื่อนไขของกฎ

กระบวนการของการเปรียบเทียบที่นิยมใช้กันมากมี 2 วิธี คือ

1. การทำดัชนี (indexing) หลักการของการแก้ปัญหาแบบนี้คือให้กฎทุกข้อที่มีส่วนของเงื่อนไขตรงกับเงื่อนไข ที่วางเอาไว้ในสถานะปัจจุบัน แล้วดึงเอากฎทุกข้อนั้นออกมา เพื่อที่จะทำการเปรียบเทียบ ถ้าหากว่าได้คำตอบ การแก้ปัญหาก็สิ้นสุด ถ้าหากว่าไม่พบคำตอบ ให้ทำการเปรียบเทียบใหม่จนพบคำตอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าดูจากเรื่องของคนโทน้ำ เราสามารถอธิบายกฎในของลักษณะของเงื่อนไขและข้อสรุปได้ดังในรูปที่ 2.4 ในส่วนของเงื่อนไขเป็นส่วนที่บอกเงื่อนไขของการใช้กฎ และในส่วนของการกระทำเป็นส่วนที่บอกที่ถึงสถานะใหม่ที่จะเป็นตัวอย่าง ดูกฎข้อที่ 1 จากตารางในรูปที่ 2.4 จะสามารถเข้าใจได้ว่า ถ้าหากในสถานะปัจจุบัน "คนโทใบที่ 1 มีน้ำน้อยกว่า 4 แกลลอน แล้วให้เติมน้ำลงในคนโทใบที่ 1จนเต็ม " ผลที่ได้จะเป็น คนโทใบแรกมีน้ำ 4 แกลลอน และใบที่สองมีน้ำอยู่เท่าเดิม (4,Y)

ถ้าหากจะพิจารณาจากสถานะเริ่มต้น ที่มีน้ำในคนโททั้งสองใบเป็น 0 หรือ (0,0) แล้วพิจารณาจากเงื่อนไขตามตารางในรูปที่ 2.4 จะเห็นได้ว่ามีกฎข้อที่ 1 และ 2 เท่านั้นที่มีเงื่อนไขที่สอดคล้อง

กฎ	เงื่อนไข	การกระทำ
1. (X,Y : X < 4) --> (4,Y)	ใบ 1 มีน้ำน้อยกว่า 4	เติมน้ำให้เต็มใบ 1
2. (X,Y : Y < 3) --> (X,3)	ใบ 2 มีน้ำน้อยกว่า 3	เติมน้ำให้เต็มใบ 2
3. (X,Y : X > 0) --> (X-D,Y)	มีน้ำอยู่ในคนโท 1	เทน้ำบางส่วนออกจาก 1
4. (X,Y : Y > 0) --> (X,Y-D)	มีน้ำอยู่ในคนโท 2	เทน้ำบางส่วนออกจาก 2
5. (X,Y : X > 0) --> (0,Y)	มีน้ำอยู่ในคนโท 1	เทน้ำออกจาก 1 จนหมด
6. (X,Y : Y > 0) --> (X,0)	มีน้ำอยู่ในคนโท 2	เทน้ำออกจาก 2 จนหมด
7. (X,Y : X+Y >= 4^Y > 0) --> (4,Y-(4-X))	มีน้ำรวม >= 4 และ คนโทใบ 2 มีน้ำอยู่	เทน้ำจากใบ 2 จนเต็มใบ 1
8. (X,Y : X+Y >= 3^X > 0) --> (X-(3-Y),3)	มีน้ำรวม >= 3 และ คนโทใบ 1 มีน้ำอยู่	เทน้ำจากใบ 1 จนเต็มใบ 2
9. (X,Y : X+Y <= 4^Y > 0) --> (X+Y,0)	มีน้ำรวม <= 4 และ คนโทใบ 2 มีน้ำอยู่	เทน้ำทั้งหมดจากใบ 2 มาใบ 1
10. (X,Y : X+Y <= 3^X > 0) --> (0,X+Y)	มีน้ำรวม <= 3 และ คนโทใบ 1 มีน้ำอยู่	เทน้ำทั้งหมดจากใบ 1 มาใบ 2

รูปที่ 2.4 ตารางแสดงเงื่อนไข (condition) และการกระทำ (action)

เมื่อนำเอากฎข้อที่ 1 มาใช้ สถานะจะเปลี่ยนจาก (0,0) เป็น (4,0)

เมื่อนำเอากฎข้อที่ 2 มาใช้ สถานะจะเปลี่ยนจาก (0,0) เป็น (0,3)

จากสถานะ (4,0) มีกฎข้อ 2 , 5 และ 8 ที่มีเงื่อนไขสอดคล้องและสถานะต่างๆจะเปลี่ยนไป ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(4,0) -> (4,3) ด้วยการใช้กฎข้อ 2

(4,0) -> (0,0) ด้วยการใช้กฎข้อ 5

(4,0) -> (1,3) ด้วยการใช้กฎข้อ 8

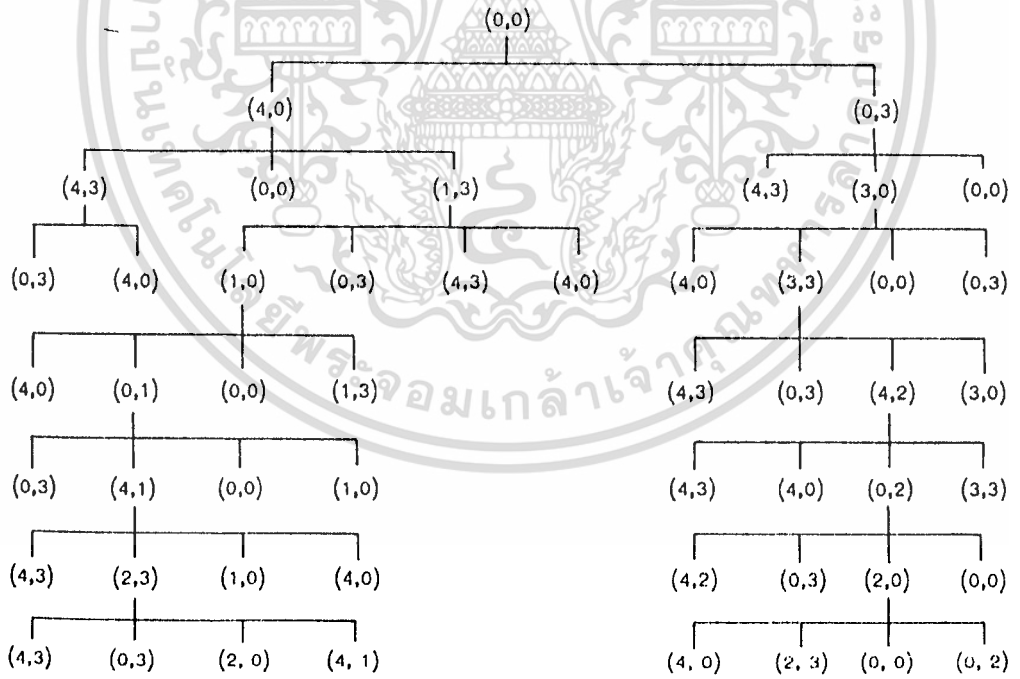
จากสถานะ (0,3) มีกฎข้อ 1,6 และ 9 ที่มีเงื่อนไขสอดคล้องสถานะต่างๆ จะเปลี่ยนไปเป็นดังนี้

(0,3) -> (4,3) ด้วยกฎข้อ 1

(0,3) -> (0,0) ด้วยกฎข้อ 6

(0,3) -> (3,0) ด้วยกฎข้อ 9

เมื่อพิจารณาเช่นนี้ไปเรื่อยๆ จะได้โครงสร้างต้นไม้ตามรูปที่ 2.5 ข้างล่างนี้



รูปที่ 2.5 รูปโครงสร้างต้นไม้ของการแก้ปัญหาแบบคนโทน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าดูจากโครงสร้างต้นไม้ที่สร้างขึ้น จะเห็นว่ามีอยู่หลายสถานะที่ไม่มีการสร้างสถานะใหม่ ทั้งนี้เนื่องจากว่าสถานะเหล่านั้นได้เกิดขึ้นมาแล้วก่อนหน้านั้นในระดับที่สูงกว่า เหตุที่ทำเช่นนี้เพราะว่าถ้าหากว่าสถานะเหล่านั้นมาสร้างสถานะใหม่ จะทำให้เกิดการวนลูป (loop) ขึ้น ซึ่งก็หมายความว่าไม่มีวันพบคำตอบ

2. การเปรียบเทียบด้วยตัวแปร (matching with variable) การแก้ปัญหาแบบนี้เป็นการอาศัยการเปรียบเทียบตัวแปรของกฎและความจริง เพื่อความเข้าใจจะขอยกตัวอย่างดังต่อไปนี้

fact: แดงใหญ่เป็นบิดาของแดง

fact: แดงเป็นบิดาของแดงเล็ก

จากความเป็นจริงดังกล่าวข้างต้น ถ้าถามว่าแดงใหญ่เป็นอะไรกับแดงเล็ก คอมพิวเตอร์จะไม่สามารถบอกความสัมพันธ์นี้ได้ การที่คอมพิวเตอร์จะสามารถบอกได้จะต้องมีกฎอะไรบางอย่างที่กล่าวถึงเรื่องดังกล่าวข้างต้น เช่น

rule : ถ้า X เป็นบิดาของ Y และ
Y เป็นบิดาของ Z
ดังนั้น X เป็นปู่ของ Z

เมื่อมีความจริงและกฎดังที่ได้แสดงแล้ว ความเป็นไปได้ของการตอบคำถามจะกระจ่างขึ้น โดยการแทนค่า X คือแดงใหญ่ และ Y คือแดง และ Z คือแดงเล็ก เราจะได้ความสัมพันธ์ของแดงใหญ่และแดงเล็กว่าคือปู่ตามกฎ ซึ่งลักษณะเช่นนี้เองที่เราเรียกว่า " การอนุมาน "

การอนุมานดังที่กล่าวมามีกฎเพียงข้อเดียวเท่านั้น ซึ่งทำได้ง่ายในกรณีที่มีกฎมากขึ้นเช่นถ้าหากเพิ่มเข้าไปอีก 1 ข้อ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

rule 1: ถ้า X เป็นพี่ของ Y และ
Y เป็นพี่ของ Z
ดังนั้น X เป็นพี่ของ Z

เมื่อเป็นเช่นนี้ ปัญหาก็จะเกิดขึ้นตามมาว่า เราจะเลือกกฎข้อใดในการอนุมาน ระหว่าง rule และ rule 1 ในที่นี้ถ้าหากว่าสำรวจดูจาก fact ที่มีอยู่ จะเห็นได้ชัดเจนว่าจะต้องเลือก rule เพราะความสัมพันธ์ของ " แดงใหญ่ " " แดง " และ " แดงเล็ก " เป็น " บิดา " ไม่ใช่ " พี่ " ดังนั้นในการอนุมานส่วนที่ทำหน้าที่ในการอนุมานจะต้องตรวจสอบความสัมพันธ์ (relation " บิดา " และ " พี่ ") ของ แอตริบริว (attribute "แดงใหญ่" "แดง" และ "แดงเล็ก") ว่าเกี่ยวพันกับกฎข้อใดจึงนำกฎข้อนั้นไปอนุมาน

วิธีการแก้ปัญหาแบบนี้ จะต้องอาศัยวิธีการแสดงความรู้ที่แตกต่างจากการแสดงความรู้ในเรื่องคนโทน้ำดังที่แสดงข้างต้น เป็นการแก้ปัญหาของความรู้ที่ถูกแสดงกฎ การแสดงความรู้อีกแบบหนึ่งที่สามารถใช้ได้ดีกับวิธีการนี้คือแบบ Predicate Logic ตัวอย่างของการแสดงความรู้ดังกล่าว คือ

Facts:

SON (John,Marry) : John เป็นลูกชายของ Marry
SON (Bill,John) : Bill เป็นลูกชายของ John
SON (Tom,Bill) : Tom เป็นลูกชายของ Bill
SON (Joe,Bill) : Joe เป็นลูกชายของ Bill
DAUGHTER (Sue,John) : Sue เป็นลูกสาวของ John
DAUGHTER (Judy,Sue) : Judy เป็นลูกสาวของ Sue

Rule:

1. SON (x,y)^SON(y,z)-> GRANDSON (x,z) :
ถ้า x เป็นลูกชายของ y และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

y เป็นลูกชายของ z

ดังนั้น x จะเป็นหลานชายของ z

2. DAUGHTER (x,y)^SON (y,z)-> GRANDAUGHTER (x,z) :

ถ้า x เป็นลูกสาวของ y และ

y เป็นลูกสาวของ z

ดังนั้น x จะเป็นหลานของ z

3. SON (x,y)^DAUGHTER (y,z)-> GRANDSON (x,z) :

ถ้า x เป็นลูกชายของ y และ

y เป็นลูกสาวของ z

ดังนั้น x จะเป็นหลานชายของ z

การค้นหาแบบฮิวริสติก

ทางด้านปัญญาประดิษฐ์ การค้นหาคำตอบอาศัยวิธีทางฮิวริสติก ความแตกต่างของการค้นหาข้อมูลแบบธรรมดาและแบบฮิวริสติกนั้น อยู่ที่การค้นหาข้อมูลแบบธรรมดาผู้ที่ทำการค้นหาข้อมูลจะต้องตรวจสอบข้อมูลทีละตัวจนครบ แต่ฮิวริสติกจะไม่ลงไปดูข้อมูลทุกตัว วิธีการนี้จะเลือกได้คำตอบที่เหมาะสมให้กับการค้นหา ซึ่งมีข้อดีคือสามารถทำการค้นหาคำตอบจากข้อมูลที่มีขนาดใหญ่มากๆได้แต่มีข้อเสียคือคำตอบที่ได้เป็นเพียงคำตอบที่ดีเท่านั้น แต่ยังไม่แน่ใจว่าจะดีที่สุด แต่เนื่องจากว่าปัญหาในบางปัญหานั้นใหญ่มาก และเป็นไปไม่ได้ที่จะทำการค้นหาด้วยวิธีธรรมดา กระบวนการของฮิวริสติกจึงเป็นสิ่งที่จำเป็น

ในเรื่องของฮิวริสติกนั้น นอกจากจะมีการค้นหาแบบฮิวริสติกแล้ว ยังมีอีกสิ่งหนึ่งที่สำคัญ คือฮิวริสติกฟังก์ชัน (heuristic function) ซึ่งหมายถึงฟังก์ชันที่ทำหน้าที่ในการวัดขนาดของความเป็นไปได้ในการแก้ปัญหาซึ่งจะแสดงด้วยตัวเลข

วิธีการดังกล่าว จะกระทำได้ โดยการพิจารณาถึงวิธีการต่างๆที่ใช้ในการแก้ปัญหา ณ สถานที่หนึ่งว่าจะสามารถแก้ปัญหาได้ตามที่ต้องการหรือไม่ โดยกำหนดเป็นน้ำหนักที่ให้กับการแก้ปัญหาของแต่ละวิธี น้ำหนักเหล่านี้จะถูกแสดงด้วยตัวเลขที่กำกับไว้กับโหนดต่างๆ ในกระบวนการค้นหาและค่าเหล่านี้ จะเป็นตัวเลขที่ใช้ในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมาณความเป็นไปได้ว่าเส้นทางที่ผ่านโหนดนั้น จะมีความเป็นไปได้ในการนำไปสู่หนทางการแก้ปัญหาได้มากน้อยแค่ไหน

จุดประสงค์ที่แท้จริงของฮิวริสติกฟังก์ชัน ก็คือการกำกับทิศทางของกระบวนการค้นหา เพื่อให้อยู่ในทิศทางที่ได้ประโยชน์มากที่สุด โดยการบอกว่าเราควรเลือกเดินทางไหนก่อน ในกรณีที่มีเส้นทางมากกว่าหนึ่งเส้นทางต้องเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการต่างๆในการค้นหาแบบฮิวริสติก

ในตอนนี้จะขอกล่าวถึงรายละเอียดต่างๆ ของการค้นหาแบบฮิวริสติกชนิดต่างๆ เท่าที่จำเป็นในการศึกษา ตามลำดับดังต่อไปนี้

Generate-and-test การค้นหาข้อมูลแบบนี้เป็นแบบที่ง่ายที่สุด มีขั้นตอน คือ

1. สร้างคำตอบที่เป็นไปได้ทั้งหมดออกมา ในที่นี้อาจจะหมายถึงการสร้างจากสถานะเริ่มต้น ถ้าเป็นการเริ่มต้นของการค้นหาข้อมูล หรืออาจจะหมายถึงการสร้างจากสถานะใดๆ ถ้าไม่ใช่การเริ่มต้น
2. ตรวจสอบว่าคำตอบที่สร้างขึ้นมานั้นถูกต้องหรือ ไม่โดยการเปรียบเทียบโหนดหรือเส้นทางที่เลือกกับชุดของสถานะเป้าหมายที่ยอมรับได้
3. หากว่าการตรวจสอบในข้อที่ 2 พบคำตอบแล้ว ก็เป็นอันว่าการค้นหาสิ้นสุด แต่ถ้าไม่พบคำตอบ ก็ให้ย้อนกลับไปทำที่ข้อ 1 ใหม่

ในวิธีการนี้ อาจจะหาคำตอบมาแบบสุ่มก็ได้ แต่ก็ไม่ได้เป็นหลักประกันว่าจะพบคำตอบ

Hill climbing เป็นวิธีการที่ดัดแปลงมาจากวิธี generate and test ต่างกันแต่เพียงว่าคำตอบของการตรวจสอบของ generate and test นั้นจะเป็นเพียง yes หรือ no เท่านั้นแต่ของ hill climbing จะให้คำตอบมาว่าเป็นอะไรที่อยู่ใกล้ที่สุดเพิ่มอีกด้วย ประกอบด้วยขั้นตอนต่างๆดังนี้

1. สร้างคำตอบในขั้นแรกขึ้นมา แล้วดูว่าเป็นคำตอบสุดท้ายได้หรือไม่ ถ้าได้เลิก ถ้าไม่ได้ทำต่อข้อถัดไป
2. จากคำตอบนี้ ให้ใช้กฎทั้งหลายหาคำตอบใหม่ออกมา แล้วให้เป็นคำตอบขั้นแรกชุดใหม่ออกมา
3. ในแต่ละองค์ประกอบของข้อ 2 ให้ทำดังนี้
   - ตรวจสอบด้วยฟังก์ชัน ถ้าเป็นคำตอบให้ยกเลิก
   - ถ้าไม่ใช่ ให้ตรวจสอบว่าตัวใดใกล้เคียงมากที่สุด

เมื่อเทียบคำตอบกับค่าที่ได้ออกมาทุกตัวที่ทดสอบมาแล้วว่าเป็นคำตอบได้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าได้ จำค่าตอบไว้ ถ้าไม่ได้ยกเลิก

4. ใช้องค์ประกอบที่ดีที่สุดจากข้อ 3 และให้เป็นค่าตอบสมมติของตัวต่อไป ในขั้นตอนนี้จะสัมพันธ์กับการแก้ปัญหา ในห้วงปัญหาตามทิศทางที่ปรากฏ ซึ่งจะนำไปสู่ค่าตอบที่เร็วที่สุด
5. กลับไปที่ขั้นตอนที่ 2

Breadth-first search

ทุกโหนดที่อยู่ในระดับเดียวกันของต้นไม้ จะถูกตรวจสอบก่อนโหนดที่อยู่ในระดับถัดไป วิธีการของการค้นหาแบบนี้ ทำโดยเริ่มจากการสร้างโหนดลูกให้กับสถานะเริ่มต้นก่อน แล้วทำการตรวจสอบว่ามีโหนดใดที่เป็นเป้าหมายได้หรือไม่ ถ้าหากว่ามีก็เป็นอันว่าการค้นหาสิ้นสุด ถ้าไม่มีจากโหนดที่เป็นโหนดที่เป็นโหนดลูกทุกตัวดังกล่าวข้างต้นก็สร้างโหนดลูกกับโหนดเหล่านั้นแล้วทำการตรวจสอบโหนดลูกทุกตัวของโหนดเหล่านั้น ถ้าพบ เป้าหมายการค้นหาก็สิ้นสุด ถ้าไม่พบก็ให้ทำการสร้างโหนดลูกของโหนดเหล่านั้นต่อไปอีก ทำเช่นนี้ไปเรื่อยๆ จนกว่าจะพบเป้าหมาย หรือจนไม่สามารถสร้างโหนดลูกใหม่ได้อีกแล้ว

การค้นหาในลักษณะนี้จะมีข้อเสียคือ

- ใช้หน่วยความจำมาก เพราะจำนวนของโหนดในแต่ละระดับจะเพิ่มขึ้นแบบเทวีคูณ
- การทำงานจะต้องมีขั้นตอนมาก
- การทำงานที่ไม่จำเป็นและการทำงานซ้ำซ้อนจะมีมาก

Best-first search เป็นกระบวนการค้นหาข้อมูลที่ได้นำเอาข้อดีของทั้ง Depth first และ Bread first มารวมกันเป็นวิธีการเดียว โดยที่แต่ละขั้นตอนของการค้นหาในโหนดลูกนั้น Best first จะเลือกเอาโหนดที่ดีที่สุด และกระบวนการในเลือกโหนดที่ดีนี้ สามารถทำได้โดยอาศัยฮิวริสติกฟังก์ชัน ซึ่งจะให้ผลของการเลือกนี้ออกเป็นคะแนน

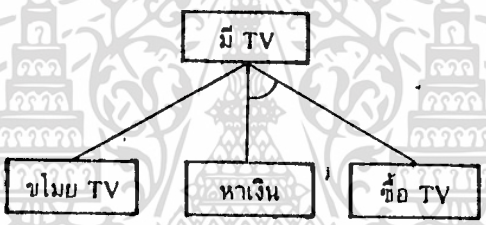
การลดทอนปัญหา (problem reduce)

การกล่าวถึงโครงสร้างต้นไม้ที่ผ่านมาเป็นเรื่องของ or กราฟเท่านั้น ซึ่งการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค้นหาบนโครงสร้างข้อมูลแบบนี้ เป็นการหาคำตอบที่มีเส้นทางเดินเพียงเส้นทางเดียว ในความเป็นจริงของการแก้ปัญหา ปัญหาบางอย่างสามารถที่จะแยกเป็นปัญหาย่อยได้ ปัญหาแบบนี้เมื่อเขียนเป็นโครงสร้างจะเรียกว่า AND-OR กราฟ รูปที่ 2.8 จะเป็นตัวอย่าง มี TV เป็นรูตโหนดที่ประกอบด้วยอาร์ค 2 อาร์ค คือขโมย TV และหาเงินซื้อ TV ซึ่งอธิบายได้ว่าการมี TV สามารถทำได้ 2 ทาง คือขโมย TV และหาเงินซื้อ TV

ในโครงสร้างของ AND-OR กราฟนี้ มีความแตกต่างจากกราฟแบบเดิมที่เป็น OR ซึ่งการเดินไปบนกิ่งนั้นสามารถจะเลือกเพียงกิ่งใดกิ่งหนึ่งเท่านั้น แต่กรณีของ AND นี้ จะต้องเลือกทุกกิ่งที่ AND กัน ดังนั้นในการ Traverse บนโครงสร้างของ AND-OR กราฟในแต่ละขั้นตอน มีสิ่งที่จะต้องทำและพิจารณาดังต่อไปนี้



รูปที่ 2.8 โครงสร้างต้นไม้ที่มีโหนดสัมพันธ์กันแบบ AND

1. การ Traverse บนกราฟเริ่มที่โหนดเริ่มต้น และเดินตามเส้นทางที่ดีที่สุดในปัจจุบัน พร้อมกับสะสมเซตของโหนดที่อยู่บนเส้นทางนั้นที่ยังไม่ได้ขยาย

2. ในการเลือกโหนดใดๆตัวหนึ่งที่ยังไม่ได้ขยายนี้ให้ขยายออก ใส่โหนดลูกในกราฟ คำนวณค่า f' สำหรับแต่ละโหนด

3. เปลี่ยนค่า f' ที่คาดคะเนเป็นค่าใหม่ตามที่ได้ขยายในขั้นตอนที่ 2 และให้เปลี่ยนตลอดทั้งกราฟ ในแต่ละโหนดที่ผ่านไปในแต่ละกราฟให้ตัดสินใจว่าอาร์คของโหนดลูกใดที่มีค่าที่ดีที่สุด แล้วบันทึกไว้ใช้เป็นส่วนหนึ่งของเส้นทางที่ดีที่สุดในระยะนี้ซึ่งอาจจะเปลี่ยนจากเดิม ในการทำเช่นนี้ จะต้องทบทวนค่าที่ได้คาดการณ์ไว้ใหม่ตามโครงสร้างต้นไม้ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การค้นหาบน AND-OR กราฟ

ในการค้นหาคำตอบบน AND-OR กราฟ มีสิ่งที่จะต้องทำความเข้าใจก่อนในเบื้องต้น ดังนี้

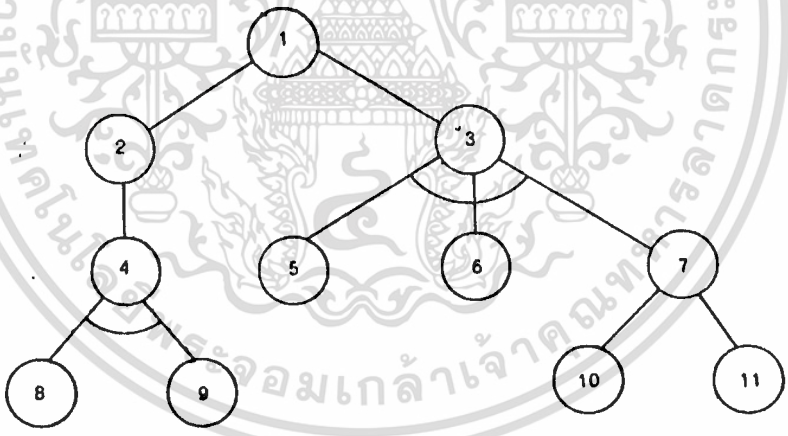
1. Solved โหนดที่ถูกพิจารณาว่าเป็น solve จะต้องมีเงื่อนไขใดเงื่อนไขหนึ่งดังต่อไปนี้

-โหนดนั้นอยู่ในเซตของเทอร์มินัล(หมายถึงโหนดสุดท้ายของโครงสร้าง)

-โหนดที่มีลูกเป็น AND และโหนดลูกทุกโหนดเป็น solved

-โหนดที่มีลูกโหนดเป็น OR และอย่างน้อยหนึ่งในโหนดลูกเหล่านั้นเป็น solve

จากรูปที่ 2.11 ถ้ากำหนดว่าโหนด 8,9,5,6,10 และ 11 เป็นเทอร์มินัลโหนด ดังนั้นคำตอบของกราฟนี้จะมีอยู่ 3 อย่าง คือ ทางเดิน {1,2,4,8,9} , {1,3,5,6,7,10} และ {1,3,5,6,7,11}



รูปที่ 2.11 AND-OR กราฟ

2. Unsolvable โหนดที่ถูกพิจารณาว่าเป็นโหนดที่แก้ปัญหาไม่ได้ จะต้องเป็นไปตามเงื่อนไขข้อใดข้อหนึ่งดังต่อไปนี้

- โหนดนั้นไม่ใช่เทอร์มินัลโหนด และเป็นโหนดที่ไม่มีโหนดลูกเนื่องจากว่าไม่มีตัวดำเนินการใดที่สามารถทำให้เกิดโหนดลูกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โหนดนั้นมีโหนดลูกเป็น AND และอย่างน้อยหนึ่งโหนดของโหนดลูกนั้นเป็น unsolvable
- โหนดที่มีโหนดลูกเป็น OR และทั้งหมดของโหนดลูกนั้นเป็น unsolvable

จากรูปที่ 2.11โหนด 1 นั้น จะเป็นโหนดที่แก้ปัญหาไม่ได้ถ้าทุกโหนดในเซตใดเซตหนึ่งดังต่อไปนี้เป็น unsolvable : {8,5} , {8,6} , {8,10,11} , {9,5} , {9,6} และ {9,10,11}

Breadth-first สำหรับ AND-OR

1.ให้โหนดเริ่มต้นอยู่ในรายการของ open ซึ่งหมายถึงโหนดที่ยังไม่ได้แตกกิ่ง

2.นำโหนดออกจาก open แล้วให้ชื่อ n

3.แตกกิ่งกับโหนด n แล้วให้ชื่อว่า m ถ้า m เป็นเซตที่มีปัญหาย่อยมากกว่าหนึ่ง ให้สร้างโหนดลูกของ m ตามลักษณะของปัญหาย่อยนั้นกับโหนดที่ถูกสร้างขึ้นมา ให้สร้างพอยน์เตอร์ชี้กลับไปยังโหนดแม่ของมันและให้จัดโหนดใหม่ทั้งหมดให้อยู่ในเซตของโหนดที่ยังไม่ได้แตกกิ่งในตอนท้ายของรายการใน open

4.ในขั้นตอนที่ 3 ถ้าหากไม่มีโหนดลูกที่ถูกสร้างขึ้น ให้ทำตามขั้นตอนนี้

4.1 ให้โหนด n เป็น unsolvable

4.2 ถ้าโหนด n ที่เป็น unsolvable ใดทำให้โหนดแม่เป็น unsolvable ก็กำหนดให้โหนดแม่เป็น unsolvable ด้วย

4.3 ถ้าโหนดเริ่มต้นเป็น unsolvable ให้เลิกการค้นหา แล้วให้การค้นหานั้นเป็น fail

4.4 โหนดใดที่อยู่ใน open และมีโหนดแม่เป็น unsolvable ให้นำออกจาก open

5. ในขั้นตอนที่ 3 ถ้าหากว่ามีการสร้างโหนดลูกที่เป็นเทอร์มินัลโหนด ให้ทำตามขั้นตอนต่อไปนี้

5.1 ให้เทอร์มินัลโหนดทุกอันเป็น solved

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ถ้าผลลัพธ์ของเทอร์มินัลโหนดนี้ ทำให้โหนดแม่ของมันเป็นsolved ให้ทำกับโหนดแม่นี้เป็น solved

5.3 ถ้าโหนดเริ่มต้นเป็น solved ให้เลิกการค้นหาและให้การค้นหานี้สำเร็จ

5.4 นำโหนดใดโหนดหนึ่งจาก open ที่เป็น solved หรือที่มีโหนดแม่เป็น solved

6. กลับไปทำขั้นตอนที่ 2

Depht-first สำหรับ AND-OR

ในการค้นหาข้อมูลแบบนี้มีวิธีการเช่นเดียวกับ Breadth-first ทุกประการ ยกเว้นอัลกอริทึมในข้อ 3 เท่านั้นซึ่งข้อ 3 ใหม่เป็นดังนี้

สร้างโหนดลูกให้กับ n แล้วให้ชื่อ m สำหรับทุกๆ m ถ้า m เป็นเซตที่มีปัญหาย่อยมากกว่าหนึ่ง ในการสร้างโหนดลูกของ m ตามลักษณะของปัญหาย่อยกับโหนดที่เพิ่งสร้างขึ้นมาใหม่ ให้สร้างพอยเตอร์ชี้กลับไปยังโหนดแม่ของมัน และจัดโหนดใหม่ทั้งหมดให้อยู่ในเซตของโหนดที่ยังไม่ได้แตกกิ่งในตอนต้นของรายการใน open

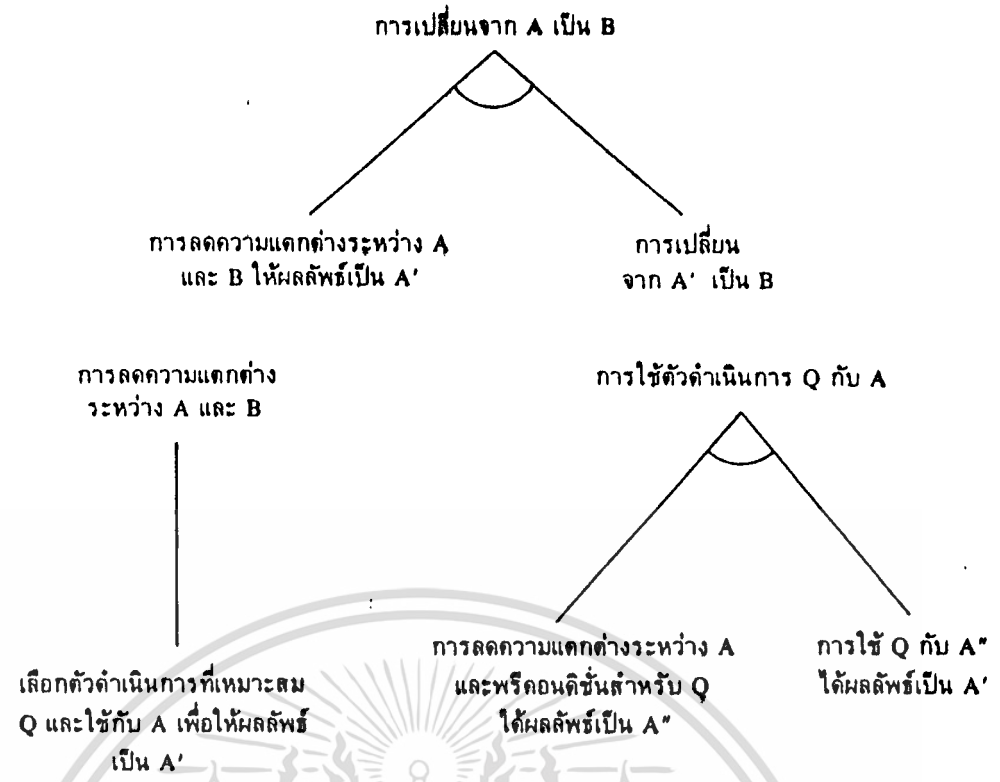
ผลลัพธ์และวิธีการ

ผลลัพธ์เป็นโครงสร้างข้อมูลที่สำคัญที่สุดของ GPS ซึ่งผลลัพธ์คือผลของการเปลี่ยนสถานะจากจุดประสงค์ปัจจุบัน ซึ่งผลลัพธ์นี้อาจจะเป็นจุดประสงค์ที่ต้องการก็ได้ ลักษณะสำคัญของผลลัพธ์ที่เกิดจากวิธีการสำคัญ 3 อย่าง คือ

1. การเปลี่ยนจากจุดประสงค์ A เป็นจุดประสงค์ B
2. การลดความแตกต่างระหว่างจุดประสงค์ A และจุดประสงค์ B โดยการดัดแปลงจุดประสงค์ A
3. การใช้ตัวดำเนินการ Q กับจุดประสงค์ A

สำหรับวิธีการต่างๆของการลดทอนปัญหาใน GSP สามารถสรุปได้ดังรูปที่ 2.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 วิธีการแก้ปัญหา 3 แบบของการลดทอนปัญหา (Barr และ Feigenbaum; 2525)

การเลือกตัวดำเนินการ

ในการพยายามที่จะเปลี่ยนจุดประสงค์ A ไปเป็นจุดประสงค์ B วิธีการของการเปลี่ยน จะใช้กระบวนการเปรียบเทียบ เพื่อที่จะหาความแตกต่างระหว่างจุดประสงค์ 2 อัน ซึ่งความแตกต่างที่เป็นไปได้ทั้งหมดนี้ จะถูกออกแบบและนำมาเรียงลำดับไว้แล้ว โดยการจัดให้อยู่ในรูปของตารางที่เชื่อมต่อกับตัวดำเนินการทั้งหลายที่สามารถลดความแตกต่างในแต่ละแบบไว้

ในระหว่างการแก้ปัญหาด้วย mean-ends analysis นี้ มีสิ่งที่พึงระวังเพื่อป้องกันการเดินทางบนทางเดินที่เราไม่ต้องการ(ผิดทาง) ดังนี้

1. ผลลัพธ์แต่ละอันจะต้องมีความแตกต่างน้อยกว่าผลลัพธ์เดิม

2. สำหรับแต่ละคู่ของโหนด AND ที่แทนผลลัพธ์ย่อยที่ถูกนำมาสร้างโดยการเปลี่ยนและการใช้นั้น ผลลัพธ์ย่อยอันที่สองจะนำมาใช้ได้ดีกว่า

3. จุดประสงค์ที่ถูกสร้างใหม่ล่าสุด ไม่ควรจะใหญ่กว่าจุดประสงค์ที่ปรากฏบนผลลัพธ์ที่อยู่สูงสุด

4. ผลลัพธ์ใดที่เคยปรากฏมาแล้วไม่ควรจะปรากฏอีกในการแก้ปัญหาแต่ละครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การแสดงความรู้ในปัญญาประดิษฐ์

การแสดงความรู้ในปัญญาประดิษฐ์

ในการแก้ปัญหาด้วยปัญญาประดิษฐ์ การแก้ปัญหาจะต้องประกอบด้วยส่วนที่สำคัญ 2 ส่วน คือส่วนที่ทำหน้าที่เกี่ยวกับการอนุมานหรือกระบวนการของการค้นหาคำตอบดังที่ได้กล่าวมาแล้ว และส่วนที่เป็นเรื่องเกี่ยวกับความรู้หรือกฎฮิวริสติกเป็นส่วนที่กระบวนการในการค้นหาคำตอบจะต้องนำไปใช้ในการอนุมาน ส่วนนี้จะประกอบด้วย กฎ ความจริง และข้ออ้างอิงต่างๆที่อธิบายเกี่ยวกับเรื่องที่ต้องการแก้ปัญหา ซึ่งผู้เชี่ยวชาญหรือบุคคลที่มีความรู้จะต้องเป็นผู้เขียนขึ้นมา การเขียนความรู้นี้จะเรียกว่าการแสดงความรู้ สำหรับการแสดงความรู้นี้ จะมีหลักการเช่นเดียวกับการเขียนโปรแกรมคอมพิวเตอร์คือผู้เชี่ยวชาญจะต้องยึดตามหลักโครงสร้างในการเขียน โครงสร้างนี้จะต้องง่ายกับผู้เขียนและคอมพิวเตอร์จะต้องเข้าใจได้ สำหรับความแตกต่างของการแสดงความรู้และการเขียนโปรแกรมที่สำคัญ คือการแสดงความรู้จะเป็นการอธิบายถึงเรื่องของกฎ ความจริง และข้ออ้างต่างๆแต่การเขียนโปรแกรมจะเป็นการอธิบายถึงกระบวนการในการแก้ปัญหาตามที่ผู้เขียนต้องการ

ในอดีตที่ผ่านมากว่า 25 ปีของวิจัยทางด้านปัญญาประดิษฐ์ ได้มีการคิดค้นวิธีการแสดงความรู้ และนำเสนอขึ้นมาหลายวิธี สามารถจำแนกวิธีการต่างๆเหล่านี้ออกเป็น 4 ประเภท คือ

1. Logical Representation Schemes คือกระบวนการของการแสดงความรู้ที่อาศัยตรรกศาสตร์ ในการแสดงความรู้แบบนี้ กฎการอนุมานและกระบวนการพิสูจน์จะใช้ความรู้ที่แสดงไว้นี้เป็นเสมือนกับปัญหาตัวอย่าง ภาษาโปรล็อกเป็นตัวอย่างของโปรแกรมที่สามารถใช้ในการแสดงความรู้แบบนี้ได้

2. Procedural Representation Schemes เป็นกระบวนการที่อาศัยการแสดงความรู้เป็นส่วนในการแก้ปัญหา ซึ่งแตกต่างจาก declarative representation ของการแสดงความรู้แบบตรรกศาสตร์และแบบอรรถศาสตร์ เช่นในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงความรู้ด้วยกฎ if...then... ในกฎสามารถที่จะแปลเป็นส่วนที่ใช้ในการแก้ปัญหาเพื่อหาเป้าหมายได้เลย ระบบการผลิตคือตัวอย่างในการแสดงความรู้แบบนี้

3. Network Representation Schemes การแสดงความรู้แบบโครงข่ายเป็นการแสดงความรู้ที่อาศัยภาพ โดยภาพของโครงข่ายนี้จะประกอบด้วยโหนดที่ใช้แทนข้อมูลข่าวสาร กฎ ความจริงและอาร์กที่ใช้แทนความสัมพันธ์ของโหนดที่เชื่อมต่อกัน 2 โหนด

4. Structure Representation Schemes การแสดงความรู้แบบนี้เป็นการขยายทีดความสามารถของการแสดงความรู้แบบโครงข่าย ซึ่งจะเป็นการช่วยทำให้โหนดแต่ละโหนดสามารถใส่ข้อมูลที่มีโครงสร้างซับซ้อนขึ้น ลักษณะของการแสดงความรู้แบบนี้คือ สคริปท์ เฟรม และออบเจ็คต์

เพรดดิเคตลอจิก (Predicate logic)

เพรดดิเคตลอจิกในบางครั้งอาจจะเรียกว่า " เพรดดิเคตแคลคูลัสหรือเฟริสท์ออร์เดอร์เพรดดิเคตลอจิก " เป็นการศึกษาทางตรรกศาสตร์ที่เกี่ยวกับการอนุมานเพื่อหาความจริงใหม่จากตรรกศาสตร์ที่มีอยู่แล้ว การอนุมานนี้กระทำบนพื้นฐานของการวิเคราะห์ประโยคทางภาษาศาสตร์โดยอาศัยหลักการทางตรรกศาสตร์แทนรูปแบบของประโยคทางภาษาศาสตร์ให้เป็นประโยคทางตรรกศาสตร์ที่เรียกว่า "ประพจน์" และประโยคทางตรรกศาสตร์นี้จะถูกวิเคราะห์ออกเป็นเทอม และเพรดดิเคต

เนื่องจากการเน้นที่การอนุมานของเพรดดิเคตนี้เอง ดังนั้นในการแสดงความรู้เพื่อการกำหนดภาษาที่เป็นตรรกศาสตร์ จึงต้องประกอบด้วยส่วนประกอบที่สำคัญ 2 ส่วนคือ สัจพจน์ และกฎการอนุมาน สัจพจน์จะเป็นเซตของถ้อยแถลง และกฎการอนุมานจะเป็นสัจพจน์ที่อธิบายถึงความจริงที่อยู่ใน formal language สำหรับการแสดงความรู้ด้วยเพรดดิเคตลอจิกนั้น จะประกอบด้วยองค์ประกอบพื้นฐานต่างๆในการแสดงความรู้คือ ชุดตัวอักษร เพรดดิเคต ตัวเชื่อม ตัวบ่งปริมาณ และภาษา

สำหรับการตรวจสอบความจริงในลอจิกนั้น เนื่องจากการแสดงความรู้ที่อาศัยเพรดดิเคตลอจิก ความรู้ต่างๆจะถูกแทนค่าด้วยลอจิกที่มีลักษณะ WFF หรือ clause ที่มีตัวแปรอยู่แล้ว การกำหนดความหมายให้กับตัวแปรจะเรียกว่าการตีความ ในลอจิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่มีอยู่จะเป็นจริงหรือเท็จ ขึ้นอยู่กับการตีความ ซึ่งลักษณะของการตีความในแบบนี้จะเรียกว่าการตรวจสอบ

โครงข่ายอรรถศาสตร์

โครงข่ายอรรถศาสตร์เป็นรูปแบบการแสดงความรู้ โดยใช้กราฟมาช่วยในการสื่อความหมาย โครงข่ายอรรถศาสตร์นี้ถูกพัฒนาขึ้นโดยข่าวสารและข้อมูลจะถูกแทนด้วยโหนดและในระหว่างโหนดจะเชื่อมต่อด้วยอาร์ก ซึ่งอาร์กแต่ละอันจะถูกกำกับด้วยป้ายชื่อที่บอกชนิดของความสัมพันธ์ของโหนด 2 อันที่เชื่อมต่อกัน

โหนด โดยปกติแล้วจะใช้แทนสิ่งของที่เป็นออบเจ็คต์และเดสคริปเตอรซึ่งออบเจ็คต์คือสิ่งที่เป็นวัตถุทางกายภาพที่สามารถมองเห็นและจับต้องได้ หรืออาจเป็นสิ่งที่เป็นลักษณะทางนิยาม หรือสิ่งที่เป็นลักษณะของนามธรรม สำหรับเดสคริปเตอร์คือสิ่งที่ขยายความเพิ่มเติมให้กับออบเจ็คต์

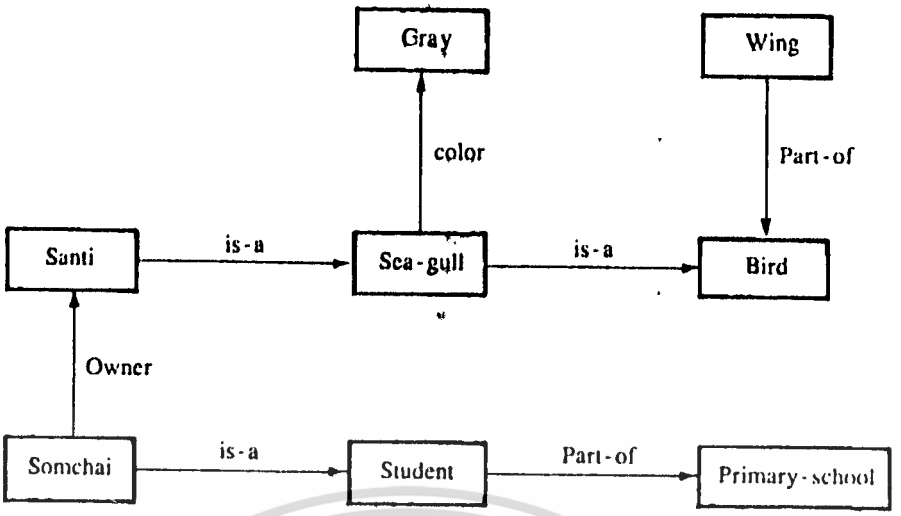
อาร์คเป็นสิ่งที่เชื่อมโหนดและเดสคริปเตอร์เข้าด้วยกัน ซึ่งอาร์คนี้จะแสดงความสัมพันธ์ของอะไรก็ได้ แต่โดยปกติแล้วมีอาร์คที่ใช้กันโดยทั่วไปคือ is-a และ has-part เป็นต้น

การหาเหตุผลด้วยวิธีอรรถศาสตร์

การหาเหตุผลของความรู้แบบนี้สามารถทำได้ง่าย โดยการค้นหาจากอาร์กของฐานความรู้นั้น แล้วทำการเปรียบเทียบดูจากโหนดที่เหมือนกัน เพื่อเปรียบเทียบหาคำตอบ หรือจะกล่าวอีกนัยหนึ่งคือ การหาเหตุผลของโครงข่ายอรรถศาสตร์ อาศัยหลักการพื้นฐานของการเปรียบเทียบแพทเทิร์น และการสืบทอดคุณสมบัติ

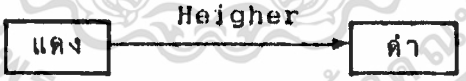
ตัวอย่างกราฟของการแสดงความรู้แบบอรรถศาสตร์ ดังในรูปที่ 3.6 ซึ่งต้องการทราบว่า SANTI มีสีอะไร ถ้าดูจากกราฟจะเห็นว่า SANTI เป็น Sea-gull มีสี Gray แต่ในการหาเหตุผล ระบบจะเริ่มดูจากอาร์ก ซึ่งในที่นี้เป็น color และ color นี้เป็นของโหนด Sea-gull แต่เนื่องจากว่า SANTI เป็นโหนดที่มีอาร์คเป็น is-a ของ Sea-gull ดังนั้นโหนดของ SANTI จึงใช้อาร์กของ color เพื่อบอกสีของ SANTI ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



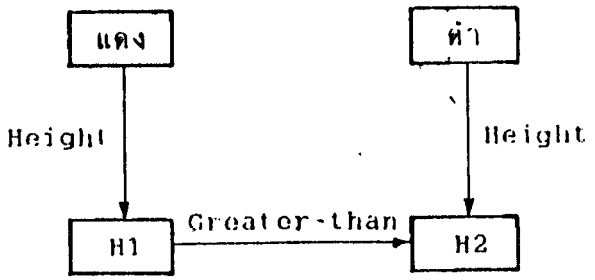
รูปที่ 3.6 กราฟของฐานความรู้แบบโครงข่ายอรรถศาสตร์

ด้วยวิธีการอนุมานความรู้ดังกล่าวข้างต้น เราจะสามารถทำการค้นหาคำตอบใดๆก็ได้ในฐานความรู้ของโครงข่ายอรรถศาสตร์ดังกล่าว ซึ่งปกติแล้วการค้นหาเหตุ-ผลของการแสดงความรู้แบบนี้ จะขึ้นอยู่กับวิธีการของการแสดงความรู้เป็นหลัก แต่เนื่องจากว่าการแสดงความรู้แบบโครงข่ายอรรถศาสตร์ดังที่ได้กล่าวมาแล้วนี้ จะมีจุดอ่อนบางอย่างเกี่ยวกับการปรับปรุงและแก้ไขความรู้ในฐานความรู้ เช่นถ้าหากเดิมมีความรู้อยู่ในโครงข่ายว่า " แดงสูงกว่าดำ " แล้วต้องการเติมว่าแดงสูง 72 นิ้ว เข้าไปในโครงข่ายจะทำได้ลำบากมาก

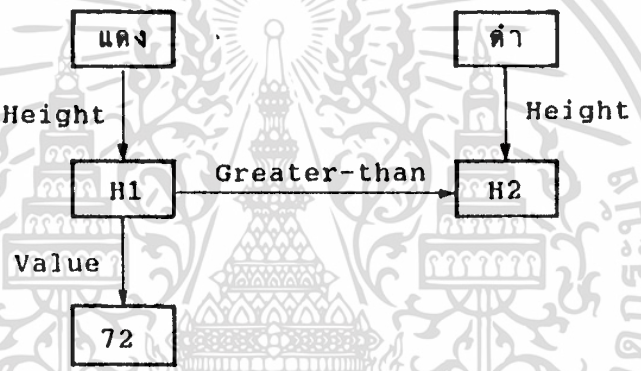


ดังนั้นในการปรับปรุงวิธีการแสดงความรู้ใหม่โดยการจำแนกโหนดออกเป็น 2 ชนิดคือ คอนเซ็พท์ และ อินสแตนท์ของคอนเซ็พท์ดังรูปข้างล่าง ซึ่งเป็นการแสดงความรู้เช่นเดียวกับรูปบน แต่การแสดงความรู้ในรูปล่างได้จำแนกโหนดออกมา ในที่นี้โหนดแดงและโหนดดำจะเป็นคอนเซ็พท์และ H1 กับ H2 จะเป็นอินสแตนท์ของโหนดแดงและดำตามลำดับ เมื่อดูตามวิธีการแสดงความรู้ดังกล่าว จะเห็นว่าอินสแตนท์ของโหนดก็คือโหนดที่แสดงความหมายบางอย่างที่ซ่อนลึกลงไปอีก ซึ่งของ H1 และ H2 จะหมายถึงอะไรบางอย่างที่บอกถึงระดับ

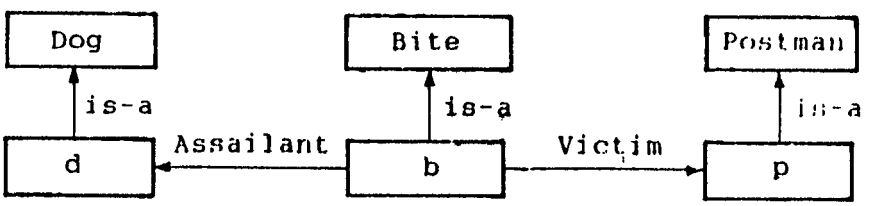
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เมื่อมีการแสดงความรู้เป็นลักษณะดังกล่าวแล้วก็จะง่ายต่อการเติมความสูงของแดงเข้าไปในฐานความรู้ ดังรูปต่อไปนี้

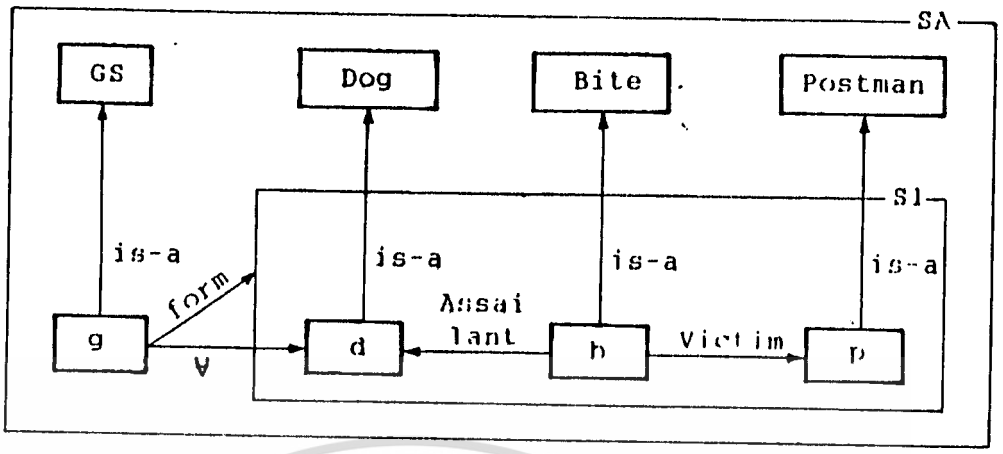


การแสดงความรู้แบบอรรถศาสตร์ด้วยวิธีการดังกล่าวนอกจากจะเป็นประโยชน์ในแง่ของการแก้ไขความรู้แล้ว ยังสามารถนำมาประยุกต์ใช้กับความรู้ที่มีลักษณะเป็นการบ่งปริมาณได้ดีด้วย ตัวอย่างต่อไปนี้จะเป็นการแสดงให้เห็นถึงวิธีการใช้อินสแตนท์ของคอนเซ็พท์



The dog bits the postman.

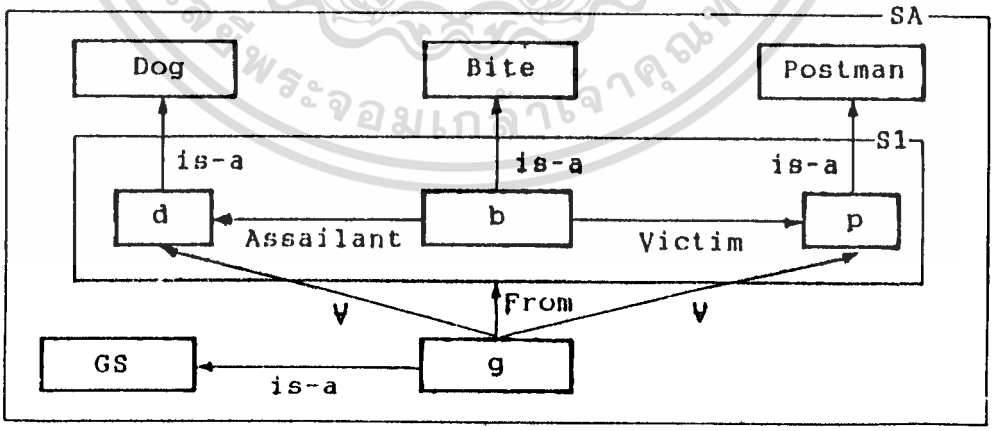
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Every dog has bitten a postman.

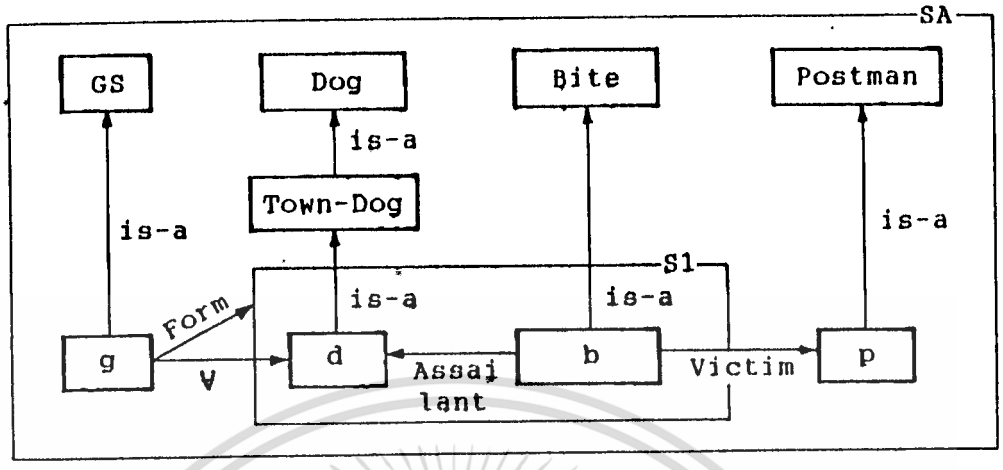
ในการแสดงความรู้ดังกล่าวข้างต้น จะเห็นว่ามีขอบเขตของตัวบ่งปริมาณของ dog อยู่ ที่จะบอกว่าเป็นสุนัขทุกตัว ในที่นี้จะต้องอาศัยโหนด g เป็นตัวบ่งบอก ซึ่ง g นี้ จะเป็นอินสแตนท์ของ GS ที่เป็นโหนดที่บอกถึงลักษณะทั่วไปของขอบเขตความรู้

โดยทั่วๆไปแล้วโหนด GS จะถูกสร้างขึ้นเสมอเมื่อมีการแสดงความรู้ที่เป็นแบบ quantification และ GS จะมีแอตทริบิวต์อย่างน้อย 2 ตัวคือ From และตัวบ่งปริมาณ for all และต่อไปนี้จะเป็นการแสดงความรู้ที่มีตัวบ่งปริมาณแบบต่างๆ



Every dog has bitten every postman

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Every dog in town has bitten the postman.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

โปรแกรมการควบคุม AGV

การทำงานของโปรแกรมจะเริ่มจาก Procedure Title จะแสดงชื่อของโปรเจ็คและผู้จัดทำ ตามด้วย Procedure Backgroundจะทำหน้าที่สร้างผังจำลองของโรงงานและเส้นทางเดินของ AGV รวมถึงโหนดต่างๆ ต่อมา Procedure Sel-Tar จะให้ผู้ใช้เลือกว่าต้องการให้มีวัตถุกีดขวางหรือไม่ เมื่อเลือกเสร็จก็จะมาเลือกเส้นทางเดินของ AGV โดยการใช้ Procedure Sel-Path ตามด้วย Procedure Path ซึ่งจะทำหน้าที่ในการกำหนดตำแหน่ง ของจุดเริ่มต้นและตำแหน่งของเป้าหมายเพื่อให้ Procedure Move เปรียบเทียบและลดความแตกต่างของตำแหน่ง การเคลื่อนที่ของ AGV นั้นจะใช้ Procedure MoUp MoDn MoRt MoLt เพื่อให้ AGV เคลื่อนที่ในแนวขึ้นบน ลงล่าง ด้านซ้าย และด้านขวาตามลำดับ โดยในแต่ละ Procedure จะมีการกำหนดตำแหน่งของ Sensor ทางด้านหน้าและด้านข้างของ AGV เมื่อมีการตรวจพบวัตถุกีดขวางก็จะได้เคลื่อนที่หลบหลีกได้

ในการเปรียบเทียบและลดความแตกต่างของตำแหน่งนั้นจะเริ่มจากการเปรียบเทียบในแนวนอนก่อน ถ้าไม่สามารถเคลื่อนที่ผ่านไปได้ ก็จะหันมาเปรียบเทียบในแนวดิ่ง และถ้ายังไม่สามารถเคลื่อนที่ได้อีก ก็จะเคลื่อนในแนวซ้ายขวาเพื่อเปรียบเทียบในแนวดิ่งหรือเคลื่อนที่ในแนวบนล่าง เพื่อจะไปเปรียบเทียบในแนวนอนต่อไป ในกรณีที่เคลื่อนที่จนครบทุกเส้นทางแล้วยังไม่สามารถผ่านไปได้ ก็จะหยุดในที่สุด ในการเคลื่อนที่นั้น ถ้าไม่สามารถผ่านไปได้ Procedure Move จะเรียกใช้ Procedure Back เพื่อให้เคลื่อนที่กลับมายังตำแหน่งเดิม เมื่อเคลื่อนที่ถึงเป้าหมาย หรือหยุดเพราะไม่สามารถเคลื่อนที่ต่อไปได้อีก Procedure Again ก็จะให้ผู้ใช้เลือกว่าจะต้องการเริ่มใหม่อีกครั้งหรือต้องการที่จะหยุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปจะเป็นการอธิบายหน้าที่ของแต่ละ procedure ดังนี้

procedure Title

procedure นี้จะทำหน้าที่แสดงชื่อโปรเจ็คบนจอภาพ โดยใช้คำสั่ง outtext xy และแสดงแบบของตัวอักษรโดยใช้คำสั่ง settextstyle

procedure Draw

precedure นี้จะทำหน้าที่สร้าง station ต่างๆของโรงงาน โดยมีพิกัดตามค่าของ Da,Db,Dc และ Dd ที่กำหนด

procedure Node

procedure นี้จะใช้แสดงตำแหน่งของจุดต่างๆภายในโรงงาน โดยแสดงเป็นตัวอักษรล้อมรอบด้วยวงกลม

procedure Background

procedure นี้จะแสดงผังของโรงงาน โดยเรียก procedure Draw มาสร้าง station ต่างๆ และเรียกใช้ procedure Node มาสร้างจุดต่างๆภายในโรงงาน

procedure GWrite

procedure นี้จะแสดงตัวอักษรต่างๆที่ป้อนจากคีย์บอร์ด ไปปรากฏบนจอภาพในโหมดกราฟิค

procedure GWriteln

procedure นี้จะขึ้นบรรทัดใหม่ ซึ่งจะแตกต่างจาก procedure GWrite

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

procedure GRO

procedure นี้จะใช้ในการแสดงตัวอักษรเฉพาะ y หรือ n ในกรณีที่ต้องการคำตอบ yes หรือ no เท่านั้น

procedure GRP

procedure นี้จะใช้แสดงตัวอักษรเฉพาะ a ถึง i เท่านั้น ซึ่งจะแสดงตำแหน่งเริ่มต้นและจุดหมายตามที่ต้องการ

procedure SelTar

procedure นี้จะแสดงทางเลือกบนจอภาพว่าต้องการให้มีวัตถุกีดขวางหรือไม่ โดยมีทางเลือก 2 ทาง คือ yes หรือ no

procedure SelPath

procedure นี้จะใช้ในการกำหนดจุดเริ่มต้นและเป้าหมาย , พร้อมทั้งกำหนดให้ค่า x และ y เป็นศูนย์

procedure Again

procedure นี้จะแสดงทางเลือกว่าต้องการ run โปรแกรมอีกครั้งหรือไม่ โดยการเลือก yes หรือ no

procedure Car

precedure นี้จะทำหน้าที่แสดงตัวของ AGV บนจอภาพ โดยมีพิกัดตามค่าของ Ca,Cb,Cc และ Cd ที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

procedure Pt

procedure นี้จะสร้าง AGV หลังจากที่ AGV ได้หยุดเรียบร้อยแล้ว โดยการเรียกใช้ procedure Car ส่วนตำแหน่งพิกัดของ AGV กำหนดได้จากค่า Ac, Bc, Cc และ Dc

procedure BaUp

procedure นี้จะใช้ในการเคลื่อนที่กลับในแนวขึ้นบนเมื่อ AGV จะชนสิ่งกีดขวาง ขณะเคลื่อนที่ในแนวบนล่าง ส่วนระยะทางในการเคลื่อนที่กำหนดจาก Dt

procedure BaDn

procedure นี้จะใช้ในการเคลื่อนที่กลับในแนวลงล่างเมื่อ AGV จะชนสิ่งกีดขวาง ขณะเคลื่อนที่ในแนวขึ้นบน ส่วนระยะทางในการเคลื่อนที่กำหนดจาก Dt

procedure BaRt

procedure นี้จะใช้ในการเคลื่อนที่กลับทางด้านขวาเมื่อ AGV จะชนสิ่งกีดขวาง ขณะเคลื่อนที่ไปทางด้านซ้าย ส่วนระยะทางในการเคลื่อนที่กำหนดจาก Dt

procedure BaLt

procedure นี้จะใช้ในการเคลื่อนที่กลับทางด้านซ้ายเมื่อ AGV จะชนสิ่งกีดขวาง ขณะเคลื่อนที่ไปทางด้านขวา ส่วนระยะทางในการเคลื่อนที่กำหนดจาก Dt

procedure Back

procedure นี้ทำหน้าที่กำหนดทิศทางในการเคลื่อนที่กลับ ว่าจะกลับในแนวใด โดยการกำหนดจากค่าของ Di ส่วนระยะทางในการเคลื่อนที่กลับนั้น จะกำหนดจากค่าของ Dt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

procedure MoUp

procedure นี้จะทำให้ AGV สามารถเคลื่อนที่ในแนวขึ้นบน โดยเมื่อพบวัตถุกีดขวางก็จะหลบหลีก แต่ถ้าไม่สามารถหลบหลีกได้ ก็จะเคลื่อนที่กลับโดยการเรียกใช้ procedure Back

procedure MoDn

procedure นี้จะทำให้ AGV สามารถเคลื่อนที่ในแนวลงล่าง โดยเมื่อพบวัตถุกีดขวางก็จะหลบหลีก แต่ถ้าไม่สามารถหลบหลีกได้ ก็จะเคลื่อนที่กลับโดยการเรียกใช้ procedure Back

procedure MoRt

procedure นี้จะทำให้ AGV สามารถเคลื่อนที่ไปทางด้านขวา โดยเมื่อพบวัตถุกีดขวางก็จะหลบหลีก แต่ถ้าไม่สามารถหลบหลีกได้ ก็จะเคลื่อนที่กลับโดยการเรียกใช้ procedure Back

procedure MoLt

procedure นี้จะทำให้ AGV สามารถเคลื่อนที่ไปทางด้านซ้าย โดยเมื่อพบวัตถุกีดขวางก็จะหลบหลีก แต่ถ้าไม่สามารถหลบหลีกได้ ก็จะเคลื่อนที่กลับโดยการเรียกใช้ procedure Back

procedure MoUn

procedure นี้จะกำหนดทิศทางในการเคลื่อนที่ของ AGV ว่าจะเคลื่อนที่ไปในทิศทางใด โดยพิจารณาจากจุดเริ่มต้นตามที่กำหนด (X1,Y1) และเป้าหมาย (X2,Y2) โดยจะพิจารณาเปรียบเทียบความแตกต่างของ X1 และ X2 แล้วเคลื่อนที่ไปในทิศทางที่จะสามารถลดความแตกต่างลงจนค่าของ X1 เท่ากับ X2 หลังจากนั้นจึงมาเปรียบเทียบความแตกต่างของ Y1 และ Y2 ต่อไป หากไม่สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เคลื่อนที่ต่อไปได้ ก็จะหันมาเปรียบเทียบความแตกต่างในแนวแกน Y ก่อนแล้วจึงไปเปรียบเทียบในแนวแกน X หากยังไม่สามารถเคลื่อนที่ไปได้อีก ก็จะมาพิจารณาว่าตำแหน่งที่ AGV อยู่นั้นมีทางเดินกี่ทาง หากมีทางเดิน 2 ทาง AGV ก็จะหยุดเดิน เนื่องจากไม่สามารถเคลื่อนที่ต่อไปได้อีก หากมีทางเดิน 3 ทาง AGV ก็จะเคลื่อนที่ไปในทางที่เหลือ เมื่อไม่สามารถเคลื่อนที่ผ่านไปได้ AGV ก็จะหยุดเดินเพราะเคลื่อนที่จนครบทุกเส้นทางแล้ว แต่ในกรณีที่มีทางเดิน 4 ทาง AGV ก็จะเคลื่อนที่ไปในทางที่เหลือ เมื่อเคลื่อนที่ผ่านไปไม่ได้ ก็จะหยุดในที่สุด procedure นี้จะใช้ในกรณีที่ X1 ไม่เท่ากับ X2 และ Y1 ไม่เท่ากับ Y2

procedure MoEq

procedure นี้จะใช้ในกรณีที่ X1 เท่ากับ X2 หรือ Y1 เท่ากับ Y2 โดยจะกำหนดทิศทางในการเคลื่อนที่ของ AGV ว่าจะเคลื่อนที่ไปในทิศทางใด การพิจารณาจะพิจารณาจากจุดเริ่มต้นตามที่กำหนด (X1,Y1) และเป้าหมาย(X2,Y2) นั่นคือจะพิจารณาเปรียบเทียบความแตกต่างของ X1 และ X2 แล้วเคลื่อนที่ไปในทิศทางที่จะสามารถลดความแตกต่างลงจนค่าของ X1 เท่ากับ X2 หลังจากนั้นจึงมาเปรียบเทียบความแตกต่างของ Y1 และ Y2 ต่อไป หากไม่สามารถเคลื่อนที่ต่อไปได้ก็จะลองเคลื่อนที่ไปในทางอื่นต่อไป หากเคลื่อนที่ผ่านไปไม่ได้และมีทางเดิน 2 ทาง AGV ก็จะหยุดเดินเนื่องจากไม่สามารถเคลื่อนที่ต่อไปได้อีก หากมีทางเดิน 3 ทาง AGV ก็จะเคลื่อนที่ไปในทางที่เหลือ เมื่อไม่สามารถเคลื่อนที่ผ่านไปได้ AGV ก็จะหยุดเดินเพราะเคลื่อนที่จนครบทุกเส้นทางแล้ว แต่ในกรณีที่มีทางเดิน 4 ทาง AGV ก็จะเคลื่อนที่ไปในทางที่เหลือ เมื่อเคลื่อนที่ผ่านไปไม่ได้ ก็จะหยุดในที่สุด

procedure Path

procedure นี้กำหนดค่า X1 และ Y1 ของจุดเริ่มต้น X2 และ Y2 ของเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

procedure Obj

procedure นี้จะทำหน้าที่สร้างวัตถุกีดขวาง โดยการกำหนดตำแหน่งของวัตถุจากค่า Oa,Ob,Oc และ Od

procedure Target

procedure นี้จะใช้ในการเคลื่อนที่วัตถุกีดขวางไปยังตำแหน่งต่างๆที่ต้องการ โดยถ้าต้องการให้วัตถุกีดขวางเคลื่อนที่ได้เร็วใช้ คีย์ Page up ในการเลื่อนวัตถุกีดขวางขึ้นบน คีย์ Page down ในการเลื่อนวัตถุกีดขวางลงล่าง คีย์ Home ในการเลื่อนวัตถุกีดขวางไปทางด้านซ้าย และคีย์ End ในการเลื่อนวัตถุกีดขวางไปทางด้านขวา แต่ถ้าต้องการให้วัตถุกีดขวางเคลื่อนที่ช้าลง จะใช้คีย์ลูกศรขึ้น ลง ซ้าย และขวาแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PROGRAM PROJECT;

USES Crt,Graph;

VAR
    GraphDriver,GraphMode,Color:integer;
    P,Q,X,Y,Z:integer;
    Tx1,Tx2,Ty1,Ty2,Px,Py:integer;
    Up,Dn,Rt,Lt,Ma,Mb,Mc,Md:integer;
    Xa,Xb,Xc,Xd,Xe,Xf,Xg,Xh:integer;
    Ya,Yb,Yc,Yd,Ye,Yf,Yg,Yh:integer;
    Ax,Bx Cx,Dx,Ex,Fx,Gx,Hx:integer;
    Ay,By,Cy,Dy,Ey,Fy,Gy,Hy:integer;
    X1,X2,Y1,Y2,Bk,Mo,Ta,Tu:integer;
    Xap,Yap,Size:word;
    Pr:Pointer;ch,key:char;Fr,Too,Tg:string;
    Trataangle:array [1..5] of PointType;

CONST
    Enter_Key = #13;  Esc_Key  = #27;
    Up_Key    = #72;  Lt_Key   = #75;
    Rt_Key    = #77;  Dn_Key   = #80;
    PgUp_Key  = #73;  PgDn_Key = #81;
    Home_Key  = #71;  End_Key  = #79;

LABEL START,STOP;

PROCEDURE Title;
begin
    SetBkColor(4);
    SetColor(14);
    SetLineStyle(0,0,3);
    SetTextStyle(1,0,8);
    SetTextJustify(1,1);
    Rectangle(0,0,Ax,Ay);
    OutTextXY(Ex,Cy,'AGV PROJECT');
    OutTextXY(Ex,Ey,'Use AI');
    SetTextStyle(1,0,2);
    OutTextXY(Ex,By,'by Mr. CHATCHAWAN WATCHARAPRAPAPONG');
    OutTextXY(Ex,Hy,'Mr.SOMCHAY ADCHARIYAVITKUL');
    Readln;
    ClearViewPort;
end;

PROCEDURE Draw(Da,Db,Dc,Dd:integer);
begin
    Trataangle[1].x:=Da;Trataangle[1].y:=Db;
    Trataangle[2].x:=Dc;Trataangle[2].y:=Db;
    Trataangle[3].x:=Dc;Trataangle[3].y:=Dd;
    Trataangle[4].x:=Da;Trataangle[4].y:=Dd;
    Trataangle[5].x:=Da;Trataangle[5].y:=Db;
    FillPoly(5,Trataangle);
end;

PROCEDURE Node(Xn,Yn:integer;Zn:string);
begin
    Circle(Xn,Yn,Py);
    OutTextXY(Xn,Yn,Zn);
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PROCEDURE Background;
begin
    SetBkColor(9);
    SetColor(15);
    SetLineStyle(0,0,1);
    SetTextStyle(0,0,1);
    Rectangle(0,0,Ax,Ay);
    Line(Xa,0,Xa,Ay);
    SetFillStyle(1,13);
    Draw(Xb,Yb,Xd,Yd);
    Draw(Xf,Yb,Xg,Yd);
    Draw(Xb,Yf,Xd,Yg);
    Draw(Xf,Yf,Xg,Yg);
    Node(Xc,Yc,'g');
    Node(Xe,Yc,'h');
    Node(Xh,Yc,'i');
    Node(Xc,Ye,'d');
    Node(Xe,Ye,'e');
    Node(Xh,Ye,'f');
    Node(Xc,Yh,'a');
    Node(Xe,Yh,'b');
    Node(Xh,Yh,'c');
    SetTextStyle(1,0,2);
    OutTextXY(Tx1,Ty2,'Station1');
    OutTextXY(Tx1,Ty1,'Station2');
    OutTextXY(Tx2,Ty1,'Station3');
    OutTextXY(Tx2,Ty2,'OFFICE');
    SetTextJustify(0,2);
    SetTextStyle(0,0,1);
    end;

PROCEDURE GWrite(var x,y:integer;gtext:string);
begin
    Outtextxy(x,y,gtext);
    x:=x+textwidth(gtext);
end;

PROCEDURE GWriteln(var x,y:integer;gtext:string);
begin
    outtextxy(x,y,gtext);
    Y:=Y+TextHeight('M');
    X:=10
end;

PROCEDURE GRO(var x,y:integer;var gtext:string);
var
    ch:char;
begin
    gtext:='';
    repeat
    ch:=readkey;
    if ch<> Enter_Key then
    if (ch='y')or(ch='Y')or(ch='n')or(ch='N') then
    begin
       GWrite(x,y,ch); gtext:=gtext+ch;
    end;
    until ch=Enter_Key
end;

PROCEDURE GRP(var x,y:integer;var gtext:string);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
var
    ch:char;
begin
    gtext:='';
    repeat
    ch:=readkey;
    if ch<> Enter_Key then
    if (ch='a') or (ch='b') or (ch='c')
    or (ch='d') or (ch='e') or (ch='f')
    or (ch='g') or (ch='h') or (ch='i ')
    or (ch='A') or (ch='B') or (ch='C')
    or (ch='D') or (ch='E') or (ch='F ')
    or (ch='G') or (ch='H') or (ch='I')
    then
    begin
       GWrite(x,y,ch); gtext:=gtext+ch;
    end;
    until ch=Enter_Key
end;

PROCEDURE Sel_Tar;
begin
    X:=10;Y:=10;
    SetViewPort(Xa+6,1,Ax-1,Ay-1,ClipOff);
    GWriteln(x,y,'Object ?');
    GWriteln(x,y,'');
    GWrite(x,y,'1)');
    SetColor(13);
    GWrite(x,y,' Y');
    SetColor(15);
    GWrite(x,y,'es');
    GWriteln(x,y,'');
    GWriteln(x,y,'');
    GWrite(x,y,'2)');
    SetColor(13);
    GWrite(x,y,' N');
    SetColor(15);
    GWrite(x,y,'o');
    GWriteln(x,y,'');
    GWriteln(x,y,'');
    GWrite(x,y,'Yours choice: ');
    SetColor(13);
    GRD(x,y,Tg);
    SetViewPort(0,0,Ax,Ay,ClipOff);
end;

PROCEDURE Sel_Path;
Begin
    X:=10;Y:=Cy;
    SetViewPort(Xa+6,1,Ax-1,Ay-1,ClipOff);
    SetColor(15);
    GWriteln(x,y,'Yours Path');
    GWriteln(x,y,'');
    GWrite(x,y,'From:');
    SetColor(13);
    GRP(x,y,Fr);
    SetColor(15);
    GWriteln(x,y,'');
    GWriteln(x,y,'');
    GWrite(x,y,'To:');
```

เอกสารนี้จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    SetColor(13);
    GRP(x,y,Too);
    SetColor(15);
    SetViewPort(0,0,Ax,Ay,ClipOff);
    X:=0;Y:=0;X1:=0;Y1:=0;X2:=0;Y2:=0;
end;

PROCEDURE Again;
begin
    X:=10;Y:=Ey;
    SetViewPort(Xa+6,1,Ax-1,Ay-1,ClipOff);
    GWriteln(x,y,'Repeat ?');
    GWriteln(x,y,'');
    GWrite(x,y,'1)');
    SetColor(13);
    GWrite(x,y,' Y');
    SetColor(15);
    GWrite(x,y,'es');
    GWriteln(x,y,'');
    GWriteln(x,y,'');
    GWrite(x,y,'2)');
    SetColor(13);
    GWrite(x,y,' N');
    SetColor(15);
    GWrite(x,y,'o');
    GWriteln(x,y,'');
    GWriteln(x,y,'');
    GWrite(x,y,'Yours choice: ');
    SetColor(13);
    GRO(x,y,Tg);
    SetViewPort(0,0,Ax,Ay,ClipOff);
end;

PROCEDURE Car(Ca,Cb,Cc,Cd:integer);
begin
    Trataangle[1].x:=Ca;Trataangle[1].y:=Cb;
    Trataangle[2].x:=Cc;Trataangle[2].y:=Cb;
    Trataangle[3].x:=Cc;Trataangle[3].y:=Cd;
    Trataangle[4].x:=Ca;Trataangle[4].y:=Cd;
    Trataangle[5].x:=Ca;Trataangle[5].y:=Cb;
    DrawPoly(5,Trataangle);
end;

PROCEDURE Pt(Pa,Pb:integer);
var Ac,Bc,Cc,Dc:integer;
begin
    Ac:=Pa-Px;Bc:=Pb-Py;Cc:=Pa+Px;Dc:=Pb+Py;
    Car(Ac,Bc,Cc,Dc);
end;

PROCEDURE BaUp(Dt:integer);
var
    Fr,Fs,Sl,Sr:integer;
begin
    Ma:=X-Px; Mb:=Y-Py; Mc:=X+Px; Md:=Y+Py;
    for Z:=1 to Dt do begin
{---------------------------------------------------------------------------}
    Fr:=Mb-2*Py-Z; Fs:=Mb-Z; Sl:=Ma-2*Px; Sr:=Mc+2*Px;

    if P=0 then Q:=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if P<0 then begin
    if (GetPixel(Sr,Md-Z)<>13)and(GetPixel(Sr,Md-Z+1)=13) then Q:=1;
    if Q=1 then begin
    if  Mc=Xb)or(Mc=Xf)or(Mc=Xa) then begin end
    else begin Ma:=Ma+1;Mc:=Mc+1;P:=P+1; end; end; end;

    if P>0 then begin
    if  GetPixel(Sl,Md-Z) <>13)and(GetPixel(Sl,Md-Z+1)=13) then Q:=1;
    if Q=1 then begin
    if  Ma=0)or(Ma=Xd)or(Ma=Xg) then begin end
    else begin Ma:=Ma-1;Mc:=Mc-1;P:=P-1; end; end; end;

    if (GetPixel(Ma,Fr)=13)
    or ((GetPixel(Ma,Fr)=13)and(GetPixel(Mc,Fr)=13))
    or ((GetPixel(Sl,Fr)=13)and(GetPixel(Ma,Fr)=13))
    or ((GetPixel(Sl,Fs)=13)and(GetPixel(Ma,Fr)=13))
    then
    begin
    if (Mc=Xb)or(Mc=Xf)or(Mc=Xa) then begin end
    else begin Ma:=Ma+1;Mc:=Mc+1;P:=P+1; end; end;

    if (GetPixel(Mc,Fr)=13)
    or ((GetPixel(Sr,Fr)=13)and(GetPixel(Mc,Fr)=13))
    or ((GetPixel(Sr,Fs)=13)and(GetPixel(Mc,Fr)=13))
    then
    begin
    if (Ma=0)or(Ma=Xd)or(Ma=Xg) then begin end
    else begin Ma:=Ma-1;Mc:=Mc-1;P:=P+1; end; end;
{-------------------------------------------------------------------------}
    Size:=ImageSize(Ma,Mb-Z,Mc,Md-Z);
    GetMem(Po,Size);
    GetImage(Ma,Mb-Z,Mc,Md-Z,Po^);
    Car(Ma,Mb-Z,Mc,Md-Z);Delay(15);
    PutImage(Ma,Mb-Z,Po^,0);
    FreeMem(Po,Size);
    end;
    X:=Ma+Px;Y:=Mb+Py-Z;
end;

PROCEDURE BaDn(Dt:integer);
var
    Fr,Fs,Sl,Sr:integer;
begin
    Ma:=X-Px; Mb:=Y-Py; Mc:=X+Px; Md:=Y+Py;
    for Z:=1 to Dt do begin
{-------------------------------------------------------------------------}
    Fr:=Md+2*Py+Z; Fs:=Md+Z; Sl:=Mc+2*Px; Sr:=Ma-2*Px;

    if P=0 then Q:=0;

    if P<0 then begin
    if (GetPixel(Sl,Mb+Z)<>13)and(GetPixel(Sl,Mb+Z-1)=13) then Q:=1;
    if Q=1 then begin
    if (Mc=Xb)or(Mc=Xf)or(Mc=Xa) then begin end
    else begin Ma:=Ma+1;Mc:=Mc+1;P:=P+1; end; end; end;

    if P>0 then begin
    if (GetPixel(Sr,Mb+Z)<>13)and(GetPixel(Sr,Mb+Z-1)=13) then Q:=1;
    if Q=1 then begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   if (Ma=0)or(Ma=Xd)or(Ma=Xg) then begin end
   else begin Ma:=Ma-1;Mc:=Mc-1;P:=P-1; end; end; end;

   if (GetPixel(Mc,Fr)=13)
   or ((GetPixel(Mc,Fr)=13)and(GetPixel(Ma,Fr)=13))
   or ((GetPixel(Sl,Fr)=13)and(GetPixel(Mc,Fr)=13))
   or ((GetPixel(Sl,Fs)=13)and(GetPixel(Mc,Fr)=13))
   then
   begin
   if (Ma=0)or(Ma=Xd)or(Ma=Xg) then begin end
   else begin Ma:=Ma-1;Mc:=Mc-1;P:=P-1; end; end;

   if (GetPixel(Ma,Fr)=13)
   or ((GetPixel(Sr,Fr)=13)and(GetPixel(Ma,Fr)=13))
   or ((GetPixel(Sr,Fs)=13)and(GetPixel(Ma,Fr)=13))
   then
   begin
   if (Mc=Xb)or(Mc=Xf)or(Mc=Xa) then begin end
   else begin Ma:=Ma+1;Mc:=Mc+1;P:=P+1; end; end;
{-------------------------------------------------------------------------------}
   Size:=ImageSize(Ma,Mb+Z,Mc,Md+Z);
   GetMem(Po,Size);
   GetImage(Ma,Mb+Z,Mc,Md+Z,Po^);
   Car(Ma,Mb+Z,Mc,Md+Z);Delay(15);
   PutImage(Ma,Mb+Z,Po^,0);
   FreeMem(Po,Size);
   end;
   X:=Ma+Px;Y:=Mb+Py+Z;
end;

PROCEDURE BaRt(Dt:integer);
var
   Fr,Fs,Sl,Sr,Mi:integer;
begin
   Ma:=X-Px; Mb:=Y-Py; Mc:=X+Px; Md:=Y+Py;
   for Z:=1 to Dt do begin
{-------------------------------------------------------------------------------}
   Fr:=Mc+2*Px+Z; Fs:=Mc+Z; Sl:=Mb-2*Py; Sr:=Md+2*Py;

   if P=0 then Q:=0;

   if P<0 then begin
   if (GetPixel(Ma+Z,Sr)<>13)and(GetPixel(Ma+Z-1,Sr)=13) then Q:=1;
   if Q=1 then begin
   if (Md=Ya)or(Md=Yb)or(Md=Yf) then begin end
   else begin Mb:=Mb+1;Md:=Md+1;P:=P+1; end; end; end;

   if P>0 then begin
   if (GetPixel(Ma+Z,Sl)<>13)and(GetPixel(Ma+Z-1,Sl)=13) then Q:=1;
   if Q=1 then begin
   if (Mb=0)or(Mb=Yd)or(Mb=Yg) then
   begin end
   else begin Mb:=Mb-1;Md:=Md-1;P:=P-1; end; end; end;

   if (GetPixel(Fr,Mb)=13)
   or ((GetPixel(Fr,Mb)=13)and(GetPixel(Fr,Md)=13))
   or ((GetPixel(Fr,Sl)=13)and(GetPixel(Fr,Mb)=13))
   or ((GetPixel(Fs,Sl)=13)and(GetPixel(Fr,Mb)=13))
   then
   begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (Md=Ya)or(Md=Yb)or(Md=Yf) then begin end
    else begin Mb:=Mb+1;Md:=Md+1;P:=P+1; end; end;

    if (GetPixel(Fr,Md)=13)
    or ((GetPixel(Fr,Sr)=13)and(GetPixel(Fr,Md)=13))
    or ((GetPixel(Fs,Sr)=13)and(GetPixel(Fr,Md)=13))
    then
    begin
    if (Mb=0)or(Mb=Yd)or(Mb=Yg) then begin end
    else begin Mb:=Mb-1;Md:=Md-1;P:=P-1; end; end;
{-------------------------------------------------------------------------}
    Size:=ImageSize(Ma+Z,Mb,Mc+Z,Md);
    GetMem(Po,Size);
    GetImage(Ma+Z,Mb,Mc+Z,Md,Po^);
    Car(Ma+Z,Mb,Mc+Z,Md);Delay(15);
    PutImage(Ma+Z,Mb,Po^,0);
    FreeMem(Po,Size);
    end;
    X:=Ma+Px+Z;Y:=Mb+Py;
end;

PROCEDURE BaLt(Dt:integer);
var
    Fr,Fs,Sl,Sr,Mi:integer;
begin
    Ma:=X-Px; Mb:=Y-Py; Mc:=X+Px; Md:=Y+Py;
    for Z:=1 to Dt do begin
{-------------------------------------------------------------------------}
    Fr:=Ma-2*Px-Z; Fs:=Ma-Z; Sl:=Md+2*Py; Sr:=Mb-2*Py;

    if P=0 then Q:=0;

    if P<0 then begin
    if (GetPixel(Mc-Z,Sl)<>13)and(GetPixel(Mc-Z+1,Sl)=13) then Q:=1;
    if Q=1 then begin
    if (Md=Ya)or(Md=Yb)or(Md=Yf) then begin end
    else begin Mb:=Mb+1;Md:=Md+1;P:=P+1; end; end; end;

    if P>0 then begin
    if (GetPixel(Mc-Z,Sr)<>13)and(GetPixel(Mc-Z+1,Sr)=13) then Q:=1;
    if Q=1 then begin
    if (Mb=0)or(Mb=Yd)or(Mb=Yg) then begin end
    else begin Mb:=Mb-1;Md:=Md-1;P:=P-1; end; end; end;

    if (GetPixel(Fr,Md)=13)
    or ((GetPixel(Fr,Md)=13)and(GetPixel(Fr,Mb)=13))
    or ((GetPixel(Fr,Sl)=13)and(GetPixel(Fr,Md)=13))
    or ((GetPixel(Fs,Sl)=13)and(GetPixel(Fr,Md)=13))
    then
    begin
    if (Mb=0)or(Mb=Yd)or(Mb=Yg) then begin end
    else begin Mb:=Mb-1;Md:=Md-1;P:=P-1; end; end;

    if (GetPixel(Fr,Mb)=13)
    or ((GetPixel(Fr,Sr)=13)and(GetPixel(Fr,Mb)=13))
    or ((GetPixel(Fs,Sr)=13)and(GetPixel(Fr,Mb)=13))
    then
    begin
    if (Md=Ya)or(Md=Yb)or(Md=Yf) then begin end
    else begin Mb:=Mb+1;Md:=Md+1;P:=P+1; end; end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{-------------------------------------------------------------------------}
   Size:=ImageSize(Ma-Z,Mb,Mc-Z,Md);
   GetMem(Po,Size);
   GetImage(Ma-Z,Mb,Mc-Z,Md,Po^);
   Car(Ma-Z,Mb,Mc-Z,Md);Delay(15);
   PutImage(Ma-Z,Mb,Po^,0);
   FreeMem(Po,Size);
   end;
   X:=Ma+Px-Z;Y:=Mb+Py;
end;

PROCEDURE Back(Dt,Di:integer);
var Cd:integer;
begin
   if Di=1 then BaUp(Dt);
   if Di=2 then BaDn(Dt);
   if Di=3 then BaRt(Dt);
   if Di=4 then BaLt(Dt);
end;

PROCEDURE MoUp(Ad:integer);
var
   Dt,Pl,Fr,Fs,Sl,Sr:integer;
begin
   if Ad=0 then Pl:=0 else Pl:=P;
   if (Ad=1)and(GetPixel(Ma,Mb-2)=13)and(GetPixel(Mc,Mb-2)=13) then
   begin
   if Rt=1 then BaRt(2*Px+1);
   if Lt=1 then BaLt(2*Px+1);
   Q:=0;
   end else
   if (Ad=1)and(GetPixel(Ma,Md+2)=13)and(GetPixel(Mc,Md+2)=13) then
   begin
   P:=0;
   end;
   Ma:=X-Px; Mb:=Y-Py; Mc:=X+Px; Md:=Y+Py;
   for Z:=1 to Yd+Pl do begin
{-------------------------------------------------------------------------
-}
   Fr:=Mb-2*Py-Z; Fs:=Mb-Z; Sl:=Ma-2*Px; Sr:=Mc+2*Px;

   if P=0 then Q:=0;

   if P<0 then begin
   if (GetPixel(Sr,Md-Z)<>13)and(GetPixel(Sr,Md-Z+1)=13) then Q:=1;
   if Q=1 then begin
   if (Mc=Xb)or(Mc=Xf)or(Mc=Xa) then begin end
   else begin Ma:=Ma+1;Mc:=Mc+1;P:=P+1; end; end; end;

   if P>0 then begin
   if (GetPixel(Sl,Md-Z)<>13)and(GetPixel(Sl,Md-Z+1)=13) then Q:=1;
   if Q=1 then begin
   if (Ma=0)or(Ma=Xd)or(Ma=Xg) then begin end
   else begin Ma:=Ma-1;Mc:=Mc-1;P:=P-1; end; end; end;

   if (GetPixel(Ma,Fr)=13)
   or ((GetPixel(Ma,Fr)=13)and(GetPixel(Mc,Fr)=13))
   or ((GetPixel(Sl,Fr)=13)and(GetPixel(Ma,Fr)=13))
   or ((GetPixel(Sl,Fs)=13)and(GetPixel(Ma,Fr)=13))
   then
   begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   if (Mc=Xb)or(Mc=Xf)or(Mc=Xa) then begin end
   else begin Ma:=Ma+1;Mc:=Mc+1;P:=P+1; end;
   end;

   if GetPixel(Ma-1,Fr)=13
   then
   begin
   if (Mc=Xb)or(Mc=Xf)or(Mc=Xa) then begin end
   else begin Ma:=Ma+1;Mc:=Mc+1;P:=P+1; end;
   end;

   if (GetPixel(Mc,Fr)=13)
   or ((GetPixel(Sr,Fr)=13)and(GetPixel(Mc,Fr)=13))
   or ((GetPixel(Sr,Fs)=13)and(GetPixel(Mc,Fr)=13))
   then
   begin
   if (Ma=0)or(Ma=Xd)or(Ma=Xg) then begin end
   else begin Ma:=Ma-1;Mc:=Mc-1;P:=P-1; end;
   end;

   if GetPixel(Mc+1,Fr)=13
   then
   begin
   if (Ma=0)or(Ma=Xd)or(Ma=Xg) then begin end
   else begin Ma:=Ma-1;Mc:=Mc-1;P:=P-1; end;
   end;

   if ((GetPixel(Ma,Fr)=13)and(GetPixel(Mc,Fr)=13)
   and(GetPixel(Sl,Fr)=13)and(GetPixel(Sr,Fr)=13))
   or (GetPixel(Ma,Mb-Z-1)=13)
   or (GetPixel(Mc,Mb-Z-1)=13)
   or (GetPixel(Ma-1,Md-Z)=13)
   or (GetPixel(Mc+1,Md-Z)=13)
   then
   begin
   X:=Ma+Px; Y:=Mb-Z+Py; Dt:=Z; Back(Dt,2); Z:=Yd+Pl; Bk:=1;
   end;
{-------------------------------------------------------------------------------
-}
   Size:=ImageSize(Ma,Mb-Z,Mc,Md-Z);
   GetMem(Po,Size);
   GetImage(Ma,Mb-Z,Mc,Md-Z,Po^);
   Car(Ma,Mb-Z,Mc,Md-Z);Delay(15);
   PutImage(Ma,Mb-Z,Po^,0);
   FreeMem(Po,Size);
   end;
   if Bk=0 then begin X:=Ma+Px;Y:=Mb+Py-Z;Y1:=Y1+1; end;
   Up:=1;Dn:=0;Rt:=0;Lt:=0;
end;

PROCEDURE MoDn(Ad:integer);
var
   Dt,Pl,Fr,Fs,Sl,Sr:integer;
begin
   if Ad=0 then Pl:=0 else Pl:=P;
   if (Ad=1)and(GetPixel(Ma,Md+2)=13)and(GetPixel(Mc,Md+2)=13) then
   begin
   if Rt=1 then BaRt(2*Px+1);
   if Lt=1 then BaLt(2*Px+1);
   G:=0;
   end else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (Ad=1)and(GetPixel(Ma,Mb-2)=13)and(GetPixel(Mc,Mb-2)=13) then
    begin
    P:=0;
    end;
    Ma:=X-Px; Mb:=Y-Py; Mc:=X+Px; Md:=Y+Py;
    for Z:=1 to Yd-Pl do begin
{-------------------------------------------------------------------------------
->
    Fr:=Md+2*Py+Z; Fs:=Md+Z; Sl:=Mc+2*Px; Sr:=Ma-2*Px;

    if P=0 then begin Q:=0; end;

    if P<0 then begin
    if (GetPixel(Sl,Mb+Z)<>13)and(GetPixel(Sl,Mb+Z-1)=13) then Q:=1;
    if Q=1 then begin
    if (Mc=Xb)or(Mc=Xf)or(Mc=Xa) then begin end
    else begin Ma:=Ma+1;Mc:=Mc+1;P:=P+1; end; end; end;

    if P>0 then begin
    if (GetPixel(Sr,Mb+Z)<>13)and(GetPixel(Sr,Mb+Z-1)=13) then Q:=1;
    if Q=1 then begin
    if (Ma=0)or(Ma=Xd)or(Ma=Xg) then begin end
    else begin Ma:=Ma-1;Mc:=Mc-1;P:=P-1; end; end; end;

    if (GetPixel(Mc,Fr)=13)
    or ((GetPixel(Mc,Fr)=13)and(GetPixel(Ma,Fr)=13))
    or ((GetPixel(Sl,Fr)=13)and(GetPixel(Mc,Fr)=13))
    or ((GetPixel(Sl,Fs)=13)and(GetPixel(Mc,Fr)=13))
    then
    begin
    if (Ma=0)or(Ma=Xd)or(Ma=Xg) then begin end
    else begin Ma:=Ma-1;Mc:=Mc-1;P:=P-1; end;
    end;

    if GetPixel(Mc+1,Fr)=13
    then
    begin
    if (Ma=0)or(Ma=Xd)or(Ma=Xg) then begin end
    else begin Ma:=Ma-1;Mc:=Mc-1;P:=P-1; end;
    end;

    if (GetPixel(Ma,Fr)=13)
    or ((GetPixel(Sl,Fr)=13)and(GetPixel(Ma,Fr)=13))
    or ((GetPixel(Sl,Fs)=13)and(GetPixel(Ma,Fr)=13))
    then
    begin
    if (Mc=Xb)or(Mc=Xf)or(Mc=Xa) then begin end
    else begin Ma:=Ma+1;Mc:=Mc+1;P:=P+1; end;
    end;

    if GetPixel(Ma-1,Fr)=13
    then
    begin
    if (Mc=Xb)or(Mc=Xf)or(Mc=Xa) then begin end
    else begin Ma:=Ma+1;Mc:=Mc+1;P:=P+1; end;
    end;

    if ((GetPixel(Mc,Fr)=13)and(GetPixel(Ma,Fr)=13)
    and(GetPixel(Sl,Fr)=13)and(GetPixel(Sr,Fr)=13))
    or (GetPixel(Ma,Md+Z+1)=13)
    or (GetPixel(Mc,Md+Z+1)=13)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    or (GetPixel(Ma-1,Mb+Z)=13)
    or (GetPixel(Mc+1,Mb+Z)=13)
    then
    begin
    X:=Ma+Px; Y:=Mb+Z+Py; Dt:=Z; Back(Dt,1); Z:=Yd-Pl; Bk:=1;
    end;
{---------------------------------------------------------------------------
-3
    Size:=ImageSize(Ma,Mb+Z,Mc,Md+Z);
    GetMem(Po,Size);
    GetImage(Ma,Mb+Z,Mc,Md+Z,Po^);
    Car(Ma,Mb+Z,Mc,Md+Z);Delay(15);
    PutImage(Ma,Mb+Z,Po^,0);
    FreeMem(Po,Size);
    end;
    if Bk=0 then begin X:=Ma+Px;Y:=Mb+Py+Z;Y1:=Y1-1; end;
    Up:=0;Dn:=1;Rt:=0;Lt:=0;
end;

PROCEDURE MoRt(Ad:integer);
var
    Dt,Pl,Fr,Fs,Sl,Sr:integer;
begin
    if Ad=0 then Pl:=0 else Pl:=P;
    if (Ad=1)and(GetPixel(Mc+2,Mb)=13)and(GetPixel(Mc+2,Md)=13) then
    begin
    if Up=1 then BaUp(2*Py+1);
    if Dn=1 then BaDn(2*Py+1);
    Q:=0;
    end else
    if (Ad=1)and(GetPixel(Ma-2,Mb)=13)and(GetPixel(Ma-2,Md)=13) then
    begin
    P:=0;
    end;
    Ma:=X-Px; Mb:=Y-Py; Mc:=X+Px; Md:=Y+Py;
    for Z:=1 to Xd-Pl do begin
{---------------------------------------------------------------------------
-3
    Fr:=Mc+2*Px+Z; Fs:=Mc+Z; Sl:=Mb-2*Py; Sr:=Md+2*Py;

    if P=0 then begin Q:=0; end;

    if P<0 then begin
    if (GetPixel(Ma+Z,Sr)<>13)and(GetPixel(Ma+Z-1,Sr)=13) then Q:=1;
    if Q=1 then begin
    if (Md=Ya)or(Md=Yb)or(Md=Yf) then begin end
    else begin Mb:=Mb+1;Md:=Md+1;P:=P+1; end; end; end;

    if P>0 then begin
    if (GetPixel(Ma+Z,Sl)<>13)and(GetPixel(Ma+Z-1,Sl)=13) then Q:=1;
    if Q=1 then begin
    if (Mb=0)or(Mb=Yd)or(Mb=Yg) then begin end
    else begin Mb:=Mb-1;Md:=Md-1;P:=P-1; end; end; end;

    if (GetPixel(Fr,Mb)=13)
    or((GetPixel(Fr,Mb)=13)and(GetPixel(Fr,Md)=13))
    or((GetPixel(Fr,Sl)=13)and(GetPixel(Fr,Mb)=13))
    or((GetPixel(Fs,Sl)=13)and(GetPixel(Fr,Mb)=13))
    then
    begin
    if (Md=Ya)or(Md=Yb)or(Md=Yf) then begin end
    else begin Mb:=Mb+1;Md:=Md+1;P:=P+1; end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    end;

    if GetPixel(Fr,Mb-1)=13
    then
    begin
    if (Md=Ya)or(Md=Yb)or(Md=Yf) then begin end
    else begin Mb:=Mb+1;Md:=Md+1;P:=P+1; end;
    end;

    if (GetPixel(Fr,Md)=13)
    or ((GetPixel(Fr,Sr)=13)and(GetPixel(Fr,Md)=13))
    or ((GetPixel(Fs,Sr)=13)and(GetPixel(Fr,Md)=13))
    then
    begin
    if (Mb=0)or(Mb=Yd)or(Mb=Yg) then begin end
    else begin Mb:=Mb-1;Md:=Md-1;P:=P-1; end;
    end;

    if GetPixel(Fr,Md+1)=13
    then
    begin
    if (Mb=0)or(Mb=Yd)or(Mb=Yg) then begin end
    else begin Mb:=Mb-1;Md:=Md-1;P:=P-1; end;
    end;

    if ((GetPixel(Fr,Mb)=13)and(GetPixel(Fr,Md)=13)
    and(GetPixel(Fr,Sl)=13)and(GetPixel(Fr,Sr)=13))
    or (GetPixel(Mc+Z+1,Mb)=13)
    or (GetPixel(Mc+Z+1,Md)=13)
    or (GetPixel(Ma+Z,Mb-1)=13)
    or (GetPixel(Ma+Z,Md+1)=13)
    then
    begin
    X:=Ma+Z+Px; Y:=Mb+Py; Dt:=Z; Back(Dt,4); Z:=Xd-P1; Bk:=1;
    end;
{-------------------------------------------------------------------------------
-}
    Size:=ImageSize(Ma+Z,Mb,Mc+Z,Md);
    GetMem(Po,Size);
    GetImage(Ma+Z,Mb,Mc+Z,Md,Po^);
    Car(Ma+Z,Mb,Mc+Z,Md);Delay(15);
    PutImage(Ma+Z,Mb,Po^,0);
    FreeMem(Po,Size);
    end;
    if Bk=0 then begin X:=Ma+Px+Z;Y:=Mb+Py;X1:=X1+1; end;
    Up:=0;Dn:=0;Rt:=1;Lt:=0;
end;

PROCEDURE MoLt(Ad:integer);
var
   Dt,P1,Fr,Fs,Sl,Sr:integer;
begin
    if Ad=0 then P1:=0 else P1:=P;
    if (Ad=1)and(GetPixel(Ma-2,Mb)=13)and(GetPixel(Ma-2,Md)=13) then
    begin
    if Up=1 then BaUp(2*Py+1);
    if Dn=1 then BaDn(2*Py+1);
    0:=0;
    end else
    if (Ad=1)and(GetPixel(Mc+2,Mb)=13)and(GetPixel(Mc+2,Md)=13) then
    begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    P:=0;
    end;
    Ma:=X-Px; Mb:=Y-Py; Mc:=X+Px; Md:=Y+Py;
    for Z:=1 to Xd+Pl do begin
{-------------------------------------------------------------------------
->
    Fr:=Ma-2*Px-Z; Fs:=Me-Z; Sl:=Md+2*Py; Sr:=Mb-2*Py;

    if P=0 then begin Q:=0; end;

    if P<0 then begin
    if (GetPixel(Mc-Z,Sl)<>13)and(GetPixel(Mc-Z+1,Sl)=13) then Q:=1;
    if Q=1 then begin
    if (Md=Ya)or(Md=Yb)or(Md=Yf) then begin end
    else begin Mb:=Mb+1;Md:=Md+1;P:=P+1; end; end; end;

    if P>0 then begin
    if (GetPixel(Mc-Z,Sr)<>13)and(GetPixel(Mc-Z+1,Sr)=13  then Q:=1;
    if Q=1 then begin
    if (Mb=0)or(Mb=Yd)or(Mb=Yg) then begin end
    else begin Mb:=Mb-1;Md:=Md-1;P:=P-1; end; end; end;

    if (GetPixel(Fr,Md)=13)
    or ((GetPixel(Fr,Md)=13)and(GetPixel(Fr,Mb)=13))
    or ((GetPixel(Fr,Sl)=13)and(GetPixel(Fr,Md)=13))
    or ((GetPixel(Fs,Sl)=13)and(GetPixel(Fr,Md)=13))
    then
    begin
    if (Mb=0)or(Mb=Yd)or(Mb=Yg) then begin end
    else begin Mb:=Mb-1;Md:=Md-1;P:=P-1; end;
    end;

    if GetPixel(Fr,Md+1)=13
    then
    begin
    if (Mb=0)or(Mb=Yd)or(Mb=Yg) then begin end
    else begin Mb:=Mb-1;Md:=Md-1;P:=P-1; end;
    end;

    if (GetPixel(Fr,Mb)=13)
    or ((GetPixel(Fr,Sr)=13)and(GetPixel(Fr,Mb)=13))
    or ((GetPixel(Fs,Sr)=13)and(GetPixel(Fr,Mb)=13))
    then
    begin
    if (Md=Ya)or(Md=Yb)or(Md=Yf) then begin end
    else begin Mb:=Mb+1;Md:=Md+1;P:=P+1; end;
    end;

    if GetPixel(Fr,Mb-1)=13
    then
    begin
    if (Md=Ya)or(Md=Yb)or(Md=Yf) then begin end
    else begin Mb:=Mb+1;Md:=Md+1;P:=P+1; end;
    end;

    if ((GetPixel(Fr,Md)=13)and(GetPixel(Fr,Mb)=13)
    and(GetPixel(Fr,Sl)=13)and(GetPixel(Fr,Sr)=13))
    or (GetPixel(Ma-Z-1,Mb)=13)
    or (GetPixel(Ma-Z-1,Md)=13)
    or (GetPixel(Mc-Z,Mb-1)=13)
    or (GetPixel(Mc-Z,Md+1)=13)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    then
    begin
    X:=Ma-Z+Px; Y:=Mb+Py; Dt:=Z; Back(Dt,3); Z:=Xd+Pl; Bk:=1;
    end;
{-------------------------------------------------------------------------------
-}
    Size:=ImageSize(Ma-Z,Mb,Mc-Z,Md);
    GetMem(Po,Size);
    GetImage(Ma-Z,Mb,Mc-Z,Md,Po^);
    Car(Ma-Z,Mb,Mc-Z,Md);Delay(15);
    PutImage(Ma-Z,Mb,Po^,0);
    FreeMem(Po,Size);
    end;
    if Bk=0 then begin X:=Ma+Px-Z;Y:=Mb+Py;X1:=X1-1; end;
    Up:=0;Dn:=0;Rt:=0;Lt:=1;
end;

PROCEDURE MoUn;
begin
    if Bk=1 then Mo:=Mo+1;
    case Mo of
      1:begin
          if X1=X2 then
          begin
            if Y1=Y2 then Pt(X,Y);
            if Y1>Y2 then begin MoDn(Tu);Tu:=0; end;
            if Y1<Y2 then begin MoUp(Tu);Tu:=0; end;
          end
          else
          begin
            if X1>X2 then begin MoLt(0);Tu:=1; end;
            if X1<X2 then begin MoRt(0);Tu:=1; end;
          end;
        end;
      2:begin
          Bk:=0;
          begin
            if Y1=Y2 then
            begin
              if X1=X2 then Pt(X,Y);
              if X1>X2 then begin MoLt(Tu);Tu:=0; end;
              if X1<X2 then begin MoRt(Tu);Tu:=0; end;
            end
            else
            begin
              if Y1>Y2 then begin MoDn(0);Tu:=1; end;
              if Y1<Y2 then begin MoUp(0);Tu:=1; end;
            end;
          end
        end;
      3:begin
          Bk:=0;
          begin
            if (X1<>2)and(Y1<>2) then begin Pt(X,Y);X1:=X2;Y1:=Y2; end;
            if X1=2 then
            begin
              if X1<X2 then begin
                MoLt(1);if Bl=1 then Mo:=3 else Mo:=2; end
                else if X1>X2 then begin
                MoRt(1);if Bk=1 then Mo:=3 else Mo:=2; end;
            end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          else
          if Y1=2 then
          begin
            if Y1<Y2 then begin
            MoDn(1);if Bk=1 then Mo:=3 else Mo:=1; end
            else if Y1>Y2 then begin
            MoUp(1);if Bk=1 then Mo:=3 else Mo:=1; end;
          end;
        end;
      end;
    4.begin
        Bk:=0;
        if ((X1=2)and(Y1<>2))or((Y1=2)and (X1<>2)) then
        begin
        Pt(X,Y);X1:=X2;Y1:=Y2;
        end;
        if  X1=2)and(Y1=2) then
        begin
        if Y1<Y2 then begin MoDn(1);Mo:=1; end
        else if Y1>Y2 then begin MoUp(1);Mo:=1; end;
        end;
      end;
  end;
end;

PROCEDURE MoEq;
begin
  if Bk=1 then Mo:=Mo+1;
  case Mo of
    1:begin
        if X1=X2 then
        begin
          if Y1=Y2 then Pt(X,Y);
          if Y1>Y2 then begin MoDn(Tu);Tu:=0; end;
          if Y1<Y2 then begin MoUp(Tu);Tu:=0; end;
        end
        else
        begin
          if Y1>X2 then begin MoLt(0);Tu:=1; end;
          if X1<X2 then begin MoRt(0);Tu:=1; end;
        end;
      end;
    2:begin
        Bk:=0;
        if (X1=X2)and(Y1<>Y2) then
        begin
          if X1=1 then begin MoRt(1); end else
          if X1=2 then begin MoRt(1); end else
          if X1=3 then begin MoLt(1); end;
          if Bk=1 then Mo:=2 else Mo:=5;
        end
        else
        if (Y1=Y2)and(X1<>X2) then
        begin
          if Y1=1 then begin MoUp(1); end else
          if Y1=2 then begin MoUp(1); end else
          if Y1=3 then begin MoDn(1); end;
          if Bk=1 then Mo:=2 else Mo:=1;
        end
      end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
3:begin
    Bk:=0;
    begin
      if (X1<>X2)and(Y1<>Y2) then begin Pt(X,Y) ;X1:=X2;Y1:=Y2; end;
      if (X1=X2)and(Y1<>Y2) th
      begin
      if X1=2 then
      begin
      MoLt(1);
      if Bk=1 then Mo:=3 else Mo:=5;
      end;
      end
      else
      if (Y1=Y2)and(X1<>X2) then
      begin
      if Y1=2 then
      begin
      MoDn(1);
      if Bk=1 then Mo:=3 else Mo:=5;
      end;
      end;
    end;
  end;
4:begin
    Bk:=0;
    if ((X1=2)and(Y1<>2))or((Y1=2)and(X1<>2)) then
    begin
    Pt(X,Y);X1:=X2;Y1:=Y2;
    end;
    if (X1=2)and(Y1=2) then
    begin
    if X1=X2 then
    begin
    if Y1<Y2 then begin MoDn(1);Mo:=2; end
    else if Y1>Y2 then begin MoUp(1);Mo:=2; end;
    end else
    if Y1=Y2 then
    begin
    if X1<X2 then begin MoLt(1);Mo:=2; end
    else if X1>X2 then begin MoRt(1);Mo:=2; end;
    end;
    end;
  end;
5:begin
    Bk:=0;
    begin
      if Y1=Y2 then
      begin
        if X1=X2 then Pt(X,Y);
        if X1>X2 then begin MoLt(Tu);Tu:=0; end;
        if X1<X2 then begin MoRt(Tu);Tu:=0; end;
      end
      else
      begin
        if Y1>Y2 then begin MoDn(0);Tu:=1; end;
        if Y1<Y2 then begin MoUp(0);Tu:=1; end;
      end;
    end
  end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;

PROCEDURE Path;
Begin

    Xc:=0;Mo:=1;P:=0;Q:=0;

    if (Fr='a')or(Fr='A') then begin X:=Xc;Y:=Yh;
    if (Too='a')or(Too='A') then begin X1:=1;Y1:=1;X2:=1;Y2:=1; end
    else if (Too='b')or(Too='B') then begin X1:=1;Y1:=1;X2:=2;Y2:=1; end
    else if (Too='c')or(Too='C') then begin X1:=1;Y1:=1;X2:=3;Y2:=1; end
    else if (Too='d')or(Too='D') then begin X1:=1;Y1:=1;X2:=1;Y2:=2; end
    else if (Too='e')or(Too='E') then begin X1:=1;Y1:=1;X2:=2;Y2:=2; end
    else if (Too='f')or(Too='F') then begin X1:=1;Y1:=1;X2:=3;Y2:=2; end
    else if (Too='g')or(Too='G') then begin X1:=1;Y1:=1;X2:=1;Y2:=3; end
    else if (Too='h')or(Too='H') ther begin X1:=1;Y1:=1;X2:=2;Y2:=3; end
    else if (Too='i')or(Too='I') then begin X1:=1;Y1:=1;X2:=3;Y2:=3; end;
    end;

    if (Fr='b')or(Fr='B')  then begin X:=Xe;Y:=Yh;
    if (Too='a')or(Too='A') then begin X1:=2;Y1:=1;X2:=1;Y2:=1; end
    else if (Too='b')or(Too='B') then begin X1:=2;Y1:=1;X2:=2;Y2:=1; end
    else if (Too='c')or(Too='C') then begin X1:=2;Y1:=1;X2:=3;Y2:=1; end
    else if (Too='d')or(Too='D') then begin X1:=2;Y1:=1;X2:=1;Y2:=2; end
    else if (Too='e')or(Too='E') then begin X1:=2;Y1:=1;X2:=2;Y2:=2; end
    else if (Too='f')or(Too='F') then begin X1:=2;Y1:=1;X2:=3;Y2:=2; end
    else if (Too='g')or(Too='G') then begin X1:=2;Y1:=1;X2:=1;Y2:=3; end
    else if (Too='h')or(Too='H') then begin X1:=2;Y1:=1;X2:=2;Y2:=3; end
    else if (Too='i')or(Too='I') then begin X1:=2;Y1:=1;X2:=3;Y2:=3; end;
    end;

    if (Fr='c')or(Fr='C') then begin X:=Xh;Y:=Yh;
    if (Too='a')or(Too='A') then begin X1:=3;Y1:=1;X2:=1;Y2:=1; end
    else if (Too='b')or(Too='B') then begin X1:=3;Y1:=1;X2:=2;Y2:=1; end
    else if (Too='c')or(Too='C') then begin X1:=3;Y1:=1;X2:=3;Y2:=1; end
    else if (Too='d')or(Too='D') then begin X1:=3;Y1:=1;X2:=1;Y2:=2; end
    else if (Too='e')or(Too='E') then begin X1:=3;Y1:=1;X2:=2;Y2:=2; end
    else if (Too='f')or(Too='F') then begin X1:=3;Y1:=1;X2:=3;Y2:=2; end
    else if (Too='g')or(Too='G') then begin X1:=3;Y1:=1;X2:=1;Y2:=3; end
    else if (Too='h')or(Too='H') then begin X1:=3;Y1:=1;X2:=2;Y2:=3; end
    else if (Too='i')or(Too='I') then begin X1:=3;Y1:=1;X2:=3;Y2:=3; end;
    end;

    if (Fr='d')or(Fr='D') then begin X:=Xc;Y:=Ye;
    if (Too='a')or(Too='A') then begin X1:=1;Y1:=2;X2:=1;Y2:=1; end
    else if (Too='b')or(Too='B') then begin X1:=1;Y1:=2;X2:=2;Y2:=1; end
    else if (Too='c')or(Too='C') then begin X1:=1;Y1:=2;X2:=3;Y2:=1; end
    else if (Too='d')or(Too='D') then begin X1:=1;Y1:=2;X2:=1;Y2:=2; end
    else if (Too='e')or(Too='E') then begin X1:=1;Y1:=2;X2:=2;Y2:=2; end
    else if (Too='f')or(Too='F') then begin X1:=1;Y1:=2;X2:=3;Y2:=2; end
    else if (Too='g')or(Too='G') then begin X1:=1;Y1:=2;X2:=1;Y2:=3; end
    else if (Too='h')or(Too='H') then begin X1:=1;Y1:=2;X2:=2;Y2:=3; end
    else if (Too='i')or(Too='I') then begin X1:=1;Y1:=2;X2:=3;Y2:=3; end;
    end;

    if (Fr='e')or(Fr='E') then begin X:=Xe;Y:=Ye;
    if (Too='a')or(Too='A') then begin X1:=2;Y1:=2;X2:=1;Y2:=1; end
    else if (Too='b')or(Too='B') then begin X1:=2;Y1:=2;X2:=2;Y2:=1; end
    else if (Too='c')or(Too='C') then begin X1:=2;Y1:=2;X2:=3;Y2:=1; end
    else if (Too='d')or(Too='D') then begin X1:=2;Y1:=2;X2:=1;Y2:=2; end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
  else if (Too='e')or(Too='E') then begin X1:=2;Y1:=2;X2:=2;Y2:=2; end
  else if (Too='f')or(Too='F') then begin X1:=2;Y1:=2;X2:=3;Y2:=2; end
  else if (Too='g')or(Too='G') then begin X1:=2;Y1:=2;X2:=1;Y2:=3; end
  else if (Too='h')or(Too='H') then begin X1:=2;Y1:=2;X2:=2;Y2:=3; end
  else if (Too='i')or(Too='I') then begin X1:=2;Y1:=2;X2:=3;Y2:=3; end;
  end;

  if (Fr='f')or(Fr='F') then begin X:=Xh;Y:=Ye;
  if (Too='a')or(Too='A') then begin X1:=3;Y1:=2;X2:=1;Y2:=1; end
  else if (Too='b')or(Too='B') then begin X1:=3;Y1:=2;X2:=2;Y2:=1; end
  else if (Too='c')or(Too='C') then begin X1:=3;Y1:=2;X2:=3;Y2:=1; end
  else if (Too='d')or(Too='D') then begin X1:=3;Y1:=2;X2:=1;Y2:=2; end
  else if (Too='e')or(Too='E') then begin X1:=3;Y1:=2;X2:=2;Y2:=2; end
  else if (Too='f')or(Too='F') then begin X1:=3;Y1:=2;X2:=3;Y2:=2; end
  else if (Too='g')or(Too='G') then begin X1:=3;Y1:=2;X2:=1;Y2:=3; end
  else if (Too='h')or(Too='H') then begin X1:=3;Y1:=2;X2:=2;Y2:=3; end
  else if (Too='i')or(Too='I') then begin X1:=3;Y1:=2;X2:=3;Y2:=3; end;
  end;

  if (Fr='g')or(Fr='G') then begin X:=Xc;Y:=Yc;
  if (Too='a')or(Too='A') then begin X1:=1;Y1:=3;X2:=1;Y2:=1; end
  else if (Too='b')or(Too='B') then begin X1:=1;Y1:=3;X2:=2;Y2:=1; end
  else if (Too='c')or(Too='C') then begin X1:=1;Y1:=3;X2:=3;Y2:=1; end
  else if (Too='d')or(Too='D') then begin X1:=1;Y1:=3;X2:=1;Y2:=2; end
  else if (Too='e')or(Too='E') then begin X1:=1;Y1:=3;X2:=2;Y2:=2; end
  else if (Too='f')or(Too='F') then begin X1:=1;Y1:=3;X2:=3;Y2:=2; end
  else if (Too='g')or(Too='G') then begin X1:=1;Y1:=3;X2:=1;Y2:=3; end
  else if (Too='h')or(Too='H') then begin X1:=1;Y1:=3;X2:=2;Y2:=3; end
  else if (Too='i')or(Too='I') then begin X1:=1;Y1:=3;X2:=3;Y2:=3; end;
  end;

  if (Fr='h')or(Fr='H') then begin X:=Xe;Y:=Yc;
  if (Too='a')or(Too='A') then begin X1:=2;Y1:=3;X2:=1;Y2:=1; end
  else if (Too='b')or(Too='B') then begin X1:=2;Y1:=3;X2:=2;Y2:=1; end
  else if (Too='c')or(Too='C') then begin X1:=2;Y1:=3;X2:=3;Y2:=1; end
  else if (Too='d')or(Too='D') then begin X1:=2;Y1:=3;X2:=1;Y2:=2; end
  else if (Too='e')or(Too='E') then begin X1:=2;Y1:=3;X2:=2;Y2:=2; end
  else if (Too='f')or(Too='F') then begin X1:=2;Y1:=3;X2:=3;Y2:=2; end
  else if (Too='g')or(Too='G') then begin X1:=2;Y1:=3;X2:=1;Y2:=3; end
  else if (Too='h')or(Too='H') then begin X1:=2;Y1:=3;X2:=2;Y2:=3; end
  else if (Too='i')or(Too='I') then begin X1:=2;Y1:=3;X2:=3;Y2:=3; end;
  end;

  if (Fr='i')or(Fr='I') then begin X:=Xh;Y:=Yc;
  if (Too='a')or(Too='A') then begin X1:=3;Y1:=3;X2:=1;Y2:=1; end
  else if (Too='b')or(Too='B') then begin X1:=3;Y1:=3;X2:=2;Y2:=1; end
  else if (Too='c')or(Too='C') then begin X1:=3;Y1:=3;X2:=3;Y2:=1; end
  else if (Too='d')or(Too='D') then begin X1:=3;Y1:=3;X2:=1;Y2:=2; end
  else if (Too='e')or(Too='E') then begin X1:=3;Y1:=3;X2:=2;Y2:=2; end
  else if (Too='f')or(Too='F') then begin X1:=3;Y1:=3;X2:=3;Y2:=2; end
  else if (Too='g')or(Too='G') then begin X1:=3;Y1:=3;X2:=1;Y2:=3; end
  else if (Too='h')or(Too='H') then begin X1:=3;Y1:=3;X2:=2;Y2:=3; end
  else if (Too='i')or(Too='I') then begin X1:=3;Y1:=3;X2:=3;Y2:=3; end
  end;
end;

PROCEDURE Obj(Oa,Ob,Oc,Od:integer);
begin
  Trataangle[1].x:=Oa;Trataangle[1].y:=Ob;
  Trataangle[2].x:=Oc;Trataangle[2].y:=Ob;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Trataangle[3].x:=Oc;Trataangle[3].y:=Od;
    Trataangle[4].x:=Oa;Trataangle[4].y:=Od;
    Trataangle[5].x:=Oa;Trataangle[5].y:=Ob;
    FillPrl (5,Trataangle);
end;

PROCEDURE Target;
var
    Oa,Ob,Oc,Od,Pst,Obn:integer;Key1,Key2:char;
begin
    For obn:=1 to 9 do begin
      Case obn of
        1:begin Oa:=Xc-Px;Ob:=Yh-Py;Oc:=Xc+Px;Od:=Yh+Py; end;
        2:begin Oa:=Xe-Px;Ob:=Yh-Py;Oc:=Xe+Px;Od:=Yh+Py; end;
        3:begin Oa:=Xh-Px;Ob:=Yh-Py;Oc:=Xh+Px;Od:=Yh+Py; end;
        4:begin Oa:=Xc-Px;Ob:=Ye-Py;Oc:=Xc+Px;Od:=Ye+Py; end;
        5:begin Oa:=Xe-Px;Ob:=Ye-Py;Oc:=Xe+Px;Od:=Ye+Py; end;
        6:begin Oa:=Xh-Px;Ob:=Ye-Py;Oc:=Xh+Px;Od:=Ye+Py; end;
        7:begin Oa:=Xc-Px;Ob:=Yc-Py;Oc:=Xc+Px;Od:=Yc+Py; end;
        8:begin Oa:=Xe-Px;Ob:=Yc-Py;Oc:=Xe+Px;Od:=Yc+Py; end;
        9:begin Oa:=Xh-Px;Ob:=Yc-Py;Oc:=Xh+Px;Od:=Yc+Py; end;
      end;
    Size:=ImageSize(Oa,Ob,Oc,Od);
    GetMem(Po,Size);
    GetImage(Oa,Ob,Oc,Od,Po^);
    Obj(Oa,Ob,Oc,Od);
    Repeat
    Key1:=Readkey;
    case Key1 of
        #0:Begin
            Key2:=ReadKey;
            case Key2 of
                Up_Key:begin
                        PutImage(Oa,Ob,Po^,0);
                        FreeMem(Po,Size);
                        if ((Oa>0) and (Oc<Xb))
                        or ((Oa>Xd) and (Oc<Xf))
                        or ((Oa>Xg) and (Oc<Xa)) then
                        begin
                        if Ob>1 then
                        begin Ob:=Ob-1; Od:=Od-1; end
                        else
                        begin Ob:=Ob;Od:=Od;Sound(1000);Delay(20);NoSound; end
                        end
                        else
                        begin
                        if (Ob>1) and (Od<Yb) then
                        begin Ob:=Ob-1; Od:=Od-1; end
                        else
                        if (Ob>Yd+1) and (Od<Yf) then
                        begin Ob:=Ob-1; Od:=Od-1; end
                        else
                        if (Ob>Yg+1) and (Od<Ya) then
                        begin Ob:=Ob-1; Od:=Od-1; end
                        else
                        begin Ob:=Ob;Od:=Od;Sound(1000);Delay(20);NoSound; end
                        end;
                        Size:=ImageSize(Oa,Ob,Oc,Od);
                        GetMem(Po,Size);
                        GetImage(Oa,Ob,Oc,Od,Po^);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Obj(Oa,Ob,Oc,Od);
        end;
Dn_Key:begin
        PutImage(Oa,Ob,Po^,0);
        FreeMem(Po,Size);
        if ((Oa>0) and (Oc<Xb))
        or ((Oa>Xd) and (Oc<Xf))
        or ((Oa>Xg) and (Oc<Xa)) then
        begin
        if Od<Ya-1 then
        begin Ob:=Ob+1; Od:=Od+1; end
        else
        begin Ob:=Ob;Od:=Od;Sound(1000);Delay(20);NoSound; end;
        end
        else
        begin
        if (Ob>0) and (Od<Yb-1) then
        begin Ob:=Ob+1; Od:=Od+1; end
        else
        if (Ob>Yd) and (Od<Yf-1) then
        begin Ob:=Ob+1; Od:=Od+1; end
        else
        if (Ob>Yg) and (Od<Ya-1) then
        begin Ob:=Ob+1; Od:=Od+1; end
        else
        begin Ob:=Ob;Od:=Od;Sound(1000);Delay(20);NoSound; end;
        end;
        Size:=ImageSize(Oa,Ob,Oc,Od);
        GetMem(Po,Size);
        GetImage(Oa,Ob,Oc,Od,Po^);
        Obj(Oa,Ob,Oc,Od);
        end;
Rt_Key:begin
        PutImage(Oa,Ob,Po^,0);
        FreeMem(Po,Size);
        if ((Ob>0) and (Od<Yb))
        or ((Ob>Yd) and (Od<Yf))
        or ((Ob>Yg) and (Od<Ya)) then
        begin
        if (Oc<Xa-1) then
        begin Oa:=Oa+1; Oc:=Oc+1; end
        else
        begin Oa:=Oa;Oc:=Oc;Sound(1000);Delay(20);NoSound; end;
        end
        else
        begin
        if (Oa>0) and (Oc<Xb-1) then
        begin Oa:=Oa+1; Oc:=Oc+1; end
        else

       if (Oa>Xd) and (Oc<Xf-1) then
       begin Oa:=Oa+1; Oc:=Oc+1; end
       else
       if (Oa>Xg) and (Oc<Xa-1) then
       begin Oa:=Oa+1; Oc:=Oc+1; end
       else
       begin Oa:=Oa;Oc:=Oc;Sound(1000);Delay(20);NoSound; end;
       end;
       Size:=ImageSize(Oa,Ob,Oc,Od);
       GetMem(Po,Size);
       GetImage(Oa,Ob,Oc,Od,Po^);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Obj(Oa,Ob,Oc,Od);
            end;
  Lt_Key:begin
            PutImage(Oa,Ob,Po^,0);
            FreeMem(Po,Size);
            if ((Ob>0) and (Od<Yb))
            or ((Ob>Yc) and (Od<Yf))
            or ((Ob>Yg) and (Od<Ya)) then
            begin
            if (Oa>1) then
            begin Oa:=Oa-1; Oc:=Oc-1; end
            else
            begin Oa:=Oa;Oc:=Oc;Sound(1000);Delay(20);NoSound; end;
            end
            else
            begin
            if (Oa>1) and (Oc<Xb) then
            begin Oa:=Oa-1; Oc:=Oc-1; end
            else
            if (Oa>Xd+1) and (Oc<Xf) then
            begin Oa:=Oa-1; Oc:=Oc-1; end
            else
            if (Oa>Xg+1) and (Oc<Xa) then
            begin Oa:=Oa-1; Oc:=Oc-1; end
            else
            begin Oa:=Oa;Oc:=Oc;Sound(1000);Delay(20);NoSound; end;
            end;
            Size:=ImageSize(Oa,Ob,Oc,Od);
            GetMem(Po,Size);
            GetImage(Oa,Ob,Oc,Od,Po^);
            Obj(Oa,Ob,Oc,Od);
            end;
PgUp_Key:begin
            PutImage(Oa,Ob,Po^,0);
            FreeMem(Po,Size);
            if ((Oa>0) and (Oc<Xb))
            or ((Oa>Xd) and (Oc<Xf))
            or ((Oa>Xg) and (Oc<Xa)) then
            begin
            if Ob>10 then
            begin Ob:=Ob-10; Od:=Od-10; end
            else
            begin Ob:=Ob;Od:=Od;Sound(1000);Delay(20);NoSound; end;
            end
            else
            begin
            if (Ob>10) and (Od<Yb) then
            begin Ob:=Ob-10; Od:=Od-10; end
            else
            if (Ob>Yd+10) and (Od<Yf) then
            begin Ob:=Ob-10; Od:=Od-10; end
            else
            if (Ob>Yg+10) and (Od<Ya) then
            begin Ob:=Ob-10; Od:=Od-10; end
            else
            begin Ob:=Ob;Od:=Od;Sound(1000);Delay(20);NoSound; end;
            end;
            Size:=ImageSize(Oa,Ob,Oc,Od);
            GetMem(Po,Size);
            GetImage(Oa,Ob,Oc,Od,Po^);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Obj(Oa,Ob,Oc,Od);
            end;
PgDn_Key:begin
            PutImage(Oa,Ob,Po^,0);
            FreeMem(Po,Size);
            if ((Oa>0) and (Oc<Xb))
            or ((Oa>Xd) and (Oc<Xf))
            or ((Oa>Xg) and (Oc<Xa)) then
            begin
            if Od<Ya-10 then
            begin Ob:=Ob+10; Od:=Od+10; end
            else
            begin Ob:=Ob;Od:=Od;Sound(1000);Delay(20);NoSound; end;
            end
            else
            begin
            if (Ob>0) and (Od<Yb-10) then
            begin Ob:=Ob+10; Od:=Od+10; end
            else
            if (Ob>Yd) and (Od<Yf-10) then
            begin Ob:=Ob+10; Od:=Od+10; end
            else
            if (Ob>Yg) and (Od<Ya-10) then
            begin Ob:=Ob+10; Od:=Od+10; end
            else
            begin Ob:=Ob;Od:=Od;Sound(1000);Delay(20);NoSound; end;
            end;
            Size:=ImageSize(Oa,Ob,Oc,Od);
            GetMem(Ro,Size);
            GetImage(Oa,Ob,Oc,Od,Po^);
            Obj(Oa,Ob,Oc,Od);
            end;
 End_Key:begin
            PutImage(Oa,Ob,Po^,0);
            FreeMem(Po,Size);
            if ((Ob>0) and (Od<Yb))
            or ((Ob>Yd) and (Od<Yf))
            or ((Ob>Yg) and (Od<Ya)) then
            begin
            if (Oc<Xa-10) then
            begin Oa:=Oa+10; Oc:=Oc+10; end
            else
            begin Oa:=Oa;Oc:=Oc;Sound(1000);Delay(20);NoSound; end;
            end
            else
            begin
            if (Oa>0) and (Oc<Xb-10) then
            begin Oa:=Oa+10; Oc:=Oc+10; end
            else
            if (Oa>Xd) and (Oc<Xf-10) then
            begin Oa:=Oa+10; Oc:=Oc+10; end
            else
            if (Oa>Xg) and (Oc<Xa-10) then
            begin Oa:=Oa+10; Oc:=Oc+10; end
            else
            begin Oa:=Oa;Oc:=Oc;Sound(1000);Delay(20);NoSound; end;
            end;
            Size:=ImageSize(Oa,Ob,Oc,Od);
            GetMem(Po,Size);
            GetImage(Oa,Ob,Oc,Od,Po^);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Obj(Oa,Ob,Oc,Od);
                    end;
          Home_Key:begin
                    PutImage(Oa,Ob,Po^,0);
                    FreeMem(Po,Size);
                    if ((Ob>0) and (Od<Yb))
                    or ((Ob>Yd) and (Oc<Yf))
                    or ((Ob>Yg) and (Od<Ya)) then
                    begin
                    if (Oa>10) then
                    begin Oa:=Oa-10; Oc:=Oc-10; end
                    else
                    begin Oa:=Oa;Oc:=Oc;Sound(1000);Delay(20);NoSound; end;
                    end
                    else
                    begin
                    if (Oa>10) and (Oc<Xb) then
                    begin Oa:=Oa-10; Oc:=Oc-10; end
                    else
                    if (Oa>Xd+10) and (Oc<Xf) then
                    begin Oa:=Oa-10; Oc:=Oc-10; end
                    else
                    if (Oa>Xg+10) and (Oc<Xa) then
                    begin Oa:=Oa-10; Oc:=Oc-10; end
                    else
                    begin Oa:=Oa;Oc:=Oc;Sound(1000);Delay(20);NoSound; end;
                    end;
                    Size:=ImageSize(Oa,Ob,Oc,Od);
                    GetMem(Po,Size);
                    GetImage(Oa,Ob,Oc,Od,Po^);
                    Obj(Oa,Ob,Oc,Od);
                    end;
          end;
          end;
      #27:begin
          Key1:=Enter_Key;
          obn:=9;
          end;
    end;
    until Key1=Enter_Key
    end;
    Sel_Path;
end;

begin {main}

    START:
    CloseGraph;
    GraphDriver:=Detect;
    InitGraph(GraphDriver,GraphMode,'');

    GetAspectRatio(Xap,Yap);

    Ax:=GetMaxX; Bx:=Round(Ax/8); Cx:=2*Bx;     Dx:=3*Bx;
    Ex:=4*Bx;    Fx:=5*Bx;        Gx:=6*Bx;     Hx:=7*Bx;
    Ay:=GetMaxY; By:=Round(Ay/8); Cy:=2*By;     Dy:=3*By;
    Ey:=4*By;    Fy:=5*By;        Gy:=6*By;     Hy:=7*By;

    Ya:=GetMaxY;                  Yb:=Round((1/7)*Ya); Yc:=Round(0.5*Yb);
    Yd:=Round(3*Yb);              Ye:=Round(3.5*Yb);   Yf:=Round(4*Yb);
```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Yg:=Round(6*Yb);              Yh:=Round(6.5*Yb);

  Xa:=Round((Yap/Xap)*Ya);  Xb:=Round((1/7)*Xa);  Xc:=Round(0.5*Xb);
  Xe:=Round(3*Xb);          Ye:=Round(3.5*Yb);    Xf:=Round(4*Xb);
  Xg:=Round(6*Xb);          Xh:=Round(6.5*Xb);

  Px:=Round((1/56)*Xa);     Py:=Round((1/56)*Ya);

  Tx1:=Round(2*Xb);     Tx2:=Round(5*Xb);
  Ty1:=Round(2*Xb);     Ty2:=Round(5*Xb);

  Title;
  Background;
  Sel_Tar;
  if (Tg='y')or(Tg='Y') then Target
  else Sel_Path;
  Path;
  if (X1<>X2)and(Y1<>Y2) then Ta:=0
  else Ta:=1;
  Ma:=X-Px;Mb:=Y-Py;Mc:=X+Px;Md:=Y+Py;
  repeat
  if Ta=0 then MoUn else
  if Ta=1 then MoEq;
  until (X1=X2)and(Y1=Y2);
  Pt(X,Y);
  Again;
  if (Tg='y')or(Tg='Y') then Goto START
  else Goto STOP;
  STOP:
  CloseGraph;

end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทสรุปและวิจารณ์

จากการทำงานของโปรแกรมที่แสดงไว้จะเห็นว่า AGV มีการเคลื่อนที่ที่สามารถหลบหลีกสิ่งกีดขวางได้ เพราะเราได้ทำการโปรแกรมเอาไว้ ดังนั้นเราสามารถที่จะนำไปใช้งานได้จริงภายในโรงงานอุตสาหกรรม หรือพื้นที่ที่เสี่ยงอันตรายสำหรับมนุษย์ ในทางความเป็นจริงแล้ว ส่วนของโปรแกรมนี้ จะต้องนำไปต่อกับส่วนโครงสร้างของตัว AGV เปรียบเสมือนการนำเอาสมองไปใส่ให้นั่นเอง นั่นคือจะต้องมีการพัฒนาในส่วนโครงสร้างของ AGV ควบคู่กันไป

การทำงานของโปรแกรมนี้ยังมีข้อบกพร่องในบางส่วน หากมีผู้ที่สนใจนำไปปรับปรุงพัฒนาอย่างจริงจังแล้ว เชื่อว่าต้องเป็นโปรแกรมที่สมบูรณ์ที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

กิตติกรรมประกาศ

โครงงาน และ ปริญญานิพนธ์นี้มีอาจที่จะสำเร็จลงได้ตามวัตถุประสงค์ถ้าขาดความร่วมมือและคำแนะนำต่างๆ ดังนั้นในนามของผู้จัดทำจึงต้องขอขอบคุณแก่ผู้ให้ความร่วมมือทุกฝ่าย อาทิ อาจารย์ที่ปรึกษาโครงงาน เพื่อนๆที่ได้ช่วยในการให้ข้อคิดเห็น ตลอดจนผู้ร่วมงานในกลุ่มที่ได้ช่วยกันแก้ปัญหา และสุดท้ายคือคุณพ่อและคุณแม่ ที่ได้กรุณาให้ความช่วยเหลือทุกด้านไม่ว่าจะเป็นเงินทุนหรือกำลังใจในการทำโครงงานนี้อย่างมาก และหวังว่าความดีที่ได้จากปริญญานิพนธ์นี้ ขอมอบกลับสู่ผู้ที่ให้ความร่วมมือและกำลังใจทุกท่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1.มงคล อัศวโกวิทกรณ์,"การเขียนโปรแกรมกราฟิคส์"

2.ภากร หุตะสังกาศ,"วิศวสารลาดกระบัง,AGV-Automated Guide Vehicle", หน้า 38-45

3.ก่อเกียรติ เก่งสกุล และ บุญเจริญ ศิริเนาวกุล, "ทฤษฎีและการประยุกต์ใช้งานปัญญาประดิษฐ์และระบบผู้เชี่ยวชาญ (Artificial Intelligence)"

4.ธนัท ชัยยุทธ และ กณพ แก้ววิชัย, "การเขียนโปรแกรมภาษาปาสคาลเบื้องต้น"

5.สุรศักดิ์ สงวนพงษ์,"เทคนิคการเขียนโปรแกรมขั้นสูง แอดวานซ์เทอร์โบปาสคาล version 4.0"

6.Eugene Charniak and Drew McDermott,"Introduction to Artificial Intelligence".

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้