

เครื่องสร้างปริภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอลทีแอลที่ใช้แอลเอสทีเอ็มที่มีการจำแนกแบบ  
หลายป้าย



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต  
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย  
ปีการศึกษา 2566

State Space Generator for LTL Model Checking using LSTM with Multi-Label  
Classification



A Dissertation Submitted in Partial Fulfillment of the Requirements  
for the Degree of Doctor of Philosophy in Computer Engineering  
Department of Computer Engineering  
Faculty Of Engineering  
Chulalongkorn University  
Academic Year 2023

หัวข้อวิทยานิพนธ์	เครื่องสร้างปริภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอลทีแอลโดยใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้าย
โดย	น.ส.ชลิกา ศักดิ์สุภาวัฒนกุล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ

---

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

.....	คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)	
คณะกรรมการสอบวิทยานิพนธ์	ประธานกรรมการ
.....	
(รองศาสตราจารย์ ดร.มชูปายาส ทองมาก)	
.....	อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ)	
.....	กรรมการ
(รองศาสตราจารย์ ดร.ดวงดาว วิชาตากุล)	
.....	กรรมการ
(รองศาสตราจารย์ ดร.ทวีชัย เสนิงวงศ์ ณ อยุธยา)	
.....	กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.พิตติพล คันธวัฒน์)	

ชลิกา ศักดิ์สุภาวัฒน์กุล : เครื่องสร้างปริภูมิสถานะสำหรับโมเดลเช็กกิงแบบแอลทีแอล  
 โดยใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้าย. ( State Space Generator for LTL  
 Model Checking using LSTM with Multi-Label Classification) อ.ที่ปรึกษาหลัก :  
 รศ. ดร.วิวัฒน์ วัฒนาวุฒิ

กระบวนการทวนสอบเชิงรูปนัยเป็นสิ่งสำคัญในการตรวจสอบความปลอดภัย ความ  
 ดำเนินชีวิตและความสัมพันธ์ของระบบซอฟต์แวร์ตั้งแต่เริ่มต้นออกแบบซอฟต์แวร์ ซึ่งสามารถช่วย  
 ให้ลดความพยายามในการพัฒนาอย่างเห็นได้ชัด โมเดลเช็กกิงเป็นวิธีการเชิงรูปนัยอย่างหนึ่ง  
 สามารถทวนสอบระบบซอฟต์แวร์ตามคุณสมบัติเชิงพฤติกรรมที่เขียนเป็นสูตรตรรกศาสตร์เชิงเวลา  
 แบบลำดับหรือเรียกว่าสูตรแอลทีแอล โดยใช้วิธีการสำรวจเส้นทางดำเนินงานที่เป็นไปได้ทั้งหมด  
 อย่างละเอียดถี่ถ้วนในปริภูมิสถานะของระบบซอฟต์แวร์และกำหนดว่าเส้นทางดำเนินงานทั้งหมด  
 ของระบบเป็นไปตามคุณสมบัติที่เขียนโดยใช้สูตรแอลทีแอลหรือไม่ อย่างไรก็ตาม ระบบที่มีขนาด  
 ใหญ่และมีความซับซ้อนอาจทำให้เกิดการระเบิดของปริภูมิสถานะในการทำโมเดลเช็กกิงได้ ซึ่งมี  
 การเสนอวิธีการลดพื้นที่ของปริภูมิสถานะหลายวิธีและแม้แต่วิธีการของแมชชีนเลิร์นนิงก็ถูก  
 นำมาใช้เพื่อทำนายผลลัพธ์ความพึงพอใจของคุณสมบัติแอลทีแอลของระบบซอฟต์แวร์ เมื่อ  
 พิจารณาถึงความสำเร็จในการฝึกอบรมโครงข่ายความจำระยะสั้นแบบยาวหรือแอลเอสทีเอ็ม  
 สำหรับการทำนายลำดับข้อมูลอนุกรมเวลาแบบยาว ในวิทยานิพนธ์นี้ได้เสนอแนวคิดแอลเอสทีเอ็ม  
 ที่ขยายการจำแนกแบบหลายป้ายและกลยุทธ์การหาขีดแบ่งเพื่อบรรเทาปัญหาการระเบิดของ  
 ปริภูมิสถานะและสร้างเป็นเครื่องสร้างปริภูมิสถานะสำหรับสร้างเส้นทางดำเนินงานของระบบ  
 ซอฟต์แวร์ได้ตามที่ต้องการ โดย ได้สร้างเครื่องมือในการทวนสอบคุณสมบัติพื้นฐานของระบบ  
 ซอฟต์แวร์ ได้แก่ คุณสมบัติด้านความปลอดภัย ด้านความดำเนินชีวิตและด้านความสัมพันธ์ ด้วย  
 เทคนิคคอน-เดอะ-ฟลาย

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ปีการศึกษา 2566

ลายมือชื่อนิสิต .....

ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

# # 6271008121 : MAJOR COMPUTER ENGINEERING

KEYWORD: Formal verification, Linear temporal logic, Long Short-Term Memory,  
multi-label classification, Model Checking

Chalika Saksupawattanakul : State Space Generator for LTL Model Checking  
using LSTM with Multi-Label Classification. Advisor: Assoc. Prof. WIWAT  
VATANAWOOD, Ph.D.

The formal verification process is important for verifying the safety, liveness and correlation of a software system from the very beginning of its software design. This obviously helps decrease development efforts. Model checking is a formal method exploited to verify a given software system against target behavioral properties written in linear temporal logic (LTL) formulas. It exhaustively explores all possible execution paths in the state space of the given software system and determines whether all execution paths satisfy the target LTL properties. However, the huge size and complexity of the software system occasionally cause the state space explosion problem in model checking. Several state space reduction methods have been proposed, and even the machine learning approach is exploited to predict the LTL property satisfaction of the software system. Considering the successful training of Long Short-Term Memory (LSTM) networks for predicting long sequences of time series data. In this dissertation, an extension of LSTM with multi-label classification and a threshold strategy is proposed to mitigate the state space explosion problem and implement the state space generator for generating execution paths of the given software system on-demand. This dissertation has developed a verification tool to verify the fundamental properties of a software system, including safety, liveness, and correlation using an on-the-fly technique.

Field of Study: Computer Engineering

Student's Signature .....

Academic Year: 2023

Advisor's Signature .....

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดีด้วยความอนุเคราะห์จาก รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ขอขอบพระคุณอาจารย์ที่ได้สละเวลาเพื่อให้คำปรึกษาสำหรับ แนวทางในการทำงานวิจัยที่ดีและคอยชี้แนะวิธีการแก้ไขปรับปรุงข้อบกพร่องในวิทยานิพนธ์นี้เพื่อให้ เกิดผลสัมฤทธิ์สูงสุดและเป็นประโยชน์ต่อการทำงานวิจัย

ขอขอบพระคุณ รองศาสตราจารย์ ดร.มยุปายาส ทองมาก ประธานกรรมการสอบภายนอก มหาวิทยาลัย รองศาสตราจารย์ ดร.ทวีติย์ เสนีวงศ์ ณ อยุธยา รองศาสตราจารย์ ดร.ดวงดาว วิชาดากุล และผู้ช่วยศาสตราจารย์ ดร.พิตติพล คັນธวัฒน์ ที่สละเวลามาให้คำแนะนำแนวทางในการทำวิทยานิพนธ์ และแก้ไขวิทยานิพนธ์ฉบับนี้ให้มีความสมบูรณ์ยิ่งขึ้น

ขอขอบคุณ อาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัยที่มอบ ความรู้ในการทำวิจัยและบุคลากรทุกท่านในภาควิชาที่ให้ความรู้ คำแนะนำและอำนวยความสะดวกใน ระหว่างที่ผู้วิจัยศึกษาอยู่ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ ตลอดจน กองทุนโครงการสร้างเสริมพลังจุฬา ฯ ก้าวสู่ศตวรรษที่ 2 (C2F) ที่ได้มอบทุนการศึกษาในการทำวิทยานิพนธ์ฉบับนี้

สุดท้ายขอบคุณครอบครัวที่คอยช่วยเหลือและเป็นกำลังใจหลักที่ช่วยผลักดันให้ทุกเรื่องผ่านพ้น ไปได้ด้วยดี เหมือนทุกครั้งที่ผ่านมา



ชลิกา ศักดิ์สุภาวัฒนกุล

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ค
บทคัดย่อภาษาอังกฤษ.....	ง
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ฉ
สารบัญตาราง.....	ญ
สารบัญรูปภาพ.....	ฎ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์.....	3
1.3 ขอบเขตการดำเนินงาน.....	4
1.4 ขั้นตอนการดำเนินงาน.....	4
1.5 ประโยชน์ที่ได้รับ.....	5
1.6 การจัดเรียงของวิทยานิพนธ์.....	5
บทที่ 2 ทฤษฎีที่เกี่ยวข้องและงานวิจัยที่เกี่ยวข้อง.....	7
2.1 ทฤษฎีที่เกี่ยวข้อง.....	7
2.1.1 ทฤษฎีที่เกี่ยวข้องกับการทวนสอบเชิงรูปนัย.....	7
2.1.1.1 โมเดลเช็กกิง (Model checking).....	7
2.1.1.2 เพทรีเน็ต (Petri net).....	9
2.1.1.3 ปริภูมิสถานะ (State space).....	12
2.1.1.4 ตรรกศาสตร์เชิงเวลาแบบลำดับ (Linear temporal logic).....	14
2.1.2 ทฤษฎีที่เกี่ยวข้องกับแบบจำลองแอลเอสทีเอ็มและการจำแนกแบบหลายป้าย.....	20

2.1.2.1	โครงข่ายความจำระยะสั้นแบบยาวหรือแอลเอสทีเอ็ม (Long short-term memory network หรือ LSTM).....	20
2.1.2.2	การจำแนกแบบหลายป้าย (Multi-label classification).....	23
2.1.2.3	กลยุทธ์การหาขีดแบ่ง (Threshold strategy).....	24
2.2	งานวิจัยที่เกี่ยวข้อง .....	26
2.2.1	งานวิจัยที่นำวิธีการแบบดั้งเดิมในการแก้ไขปัญหาการระเบิดของปริภูมิสถานะ .....	26
2.2.2	งานวิจัยที่ประยุกต์ใช้แมชชีนเลิร์นนิงในโมเดลเชิงกึ่ง .....	27
2.2.3	งานวิจัยที่นำแบบจำลองแอลเอสทีเอ็มและการจำแนกแบบหลายป้าย.....	29
บทที่ 3	แนวความคิดการพัฒนาเครื่องสร้างปริภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอลทีแอลโดยใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้าย .....	31
3.1	การจัดเตรียมข้อมูลสำหรับการเรียนรู้ของแบบจำลอง LSTM-MLC .....	32
3.1.1	การถอดรหัสการเข้าถึงมาร์กกิงทั้งหมด .....	32
3.1.2	การสกัดเส้นทางย่อยจากกราฟปริภูมิสถานะ.....	35
3.1.3	การแปลงชื่อสถานะไปเป็นรหัสโทเคน.....	36
3.1.4	การสร้างชุดข้อมูลสำหรับแบบจำลอง LSTM-MLC .....	37
3.2	การออกแบบและพัฒนาเครื่องสร้างปริภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอลทีแอลโดยใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้าย .....	41
3.2.1	การออกแบบและสร้างแบบจำลอง LSTM-MLC.....	41
3.2.2	การเรียนรู้ของแบบจำลอง LSTM-MLC.....	42
3.2.3	การกำหนดขีดแบ่งที่เหมาะสมสำหรับแบบจำลอง LSTM-MLC.....	43
3.2.4	การประเมินประสิทธิภาพการจำแนกหลายป้ายของแบบจำลอง LSTM-MLC .....	44
3.3	การสร้างปริภูมิสถานะโดยเครื่องสร้างปริภูมิสถานะ .....	45
บทที่ 4	แนวความคิดการพัฒนาเครื่องมือการทวนสอบคุณสมบัติโดยใช้เครื่องสร้างปริภูมิสถานะ ...	47
4.1	การกำหนดรูปแบบของสูตรแอลทีแอลแบบเฉพาะ .....	48
4.2	การแปลงชื่อเพลสไปยังชื่อสถานะที่สอดคล้อง .....	49



4.3	การกำหนดการตรวจสอบความถูกต้องของการระบุสูตรแอลทีแอลแบบเฉพาะ	51
4.4	การทวนสอบคุณสมบัติการทำงานของระบบจากสูตรแอลทีแอลแบบเฉพาะ	53
4.4.1	การทวนสอบคุณสมบัติด้านความปลอดภัย	53
4.4.2	การทวนสอบคุณสมบัติด้านความดำเนินชีวิต	54
4.4.3	การทวนสอบคุณสมบัติด้านความสัมพันธ์	55
4.4.4	การทวนสอบคุณสมบัติสูตรแอลทีแอลประเภท Until	57
บทที่ 5	การออกแบบสถาปัตยกรรมระบบและการพัฒนาระบบ	60
5.1	การวิเคราะห์และออกแบบระบบ	60
5.1.1	ความต้องการเชิงฟังก์ชัน	60
5.1.2	การออกแบบระบบ	61
5.2	การออกแบบสถาปัตยกรรม	63
5.3	การพัฒนาระบบ	68
บทที่ 6	การทดสอบเครื่องสร้างปริภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอลทีแอลโดยใช้แอลเอส ที่เอ็มที่มีการจำแนกแบบหลายป้าย	70
6.1	การออกแบบกรณีศึกษา	70
6.1.1	กรณีศึกษาที่ 1 แบบจำลองเพทรีเน็ตสำหรับระบบเครื่องรับจ่ายเงินอัตโนมัติ	71
6.1.2	กรณีศึกษาที่ 2 แบบจำลองเพทรีเน็ตสำหรับระบบการรับประทานอาหารของ นักปราชญ์ 9 คน	75
6.1.3	กรณีศึกษาที่ 3 แบบจำลองเพทรีเน็ตสำหรับระบบสัญญาณไฟจราจรสี่ต้น	78
6.1.4	กรณีศึกษาที่ 4 แบบจำลองเพทรีเน็ตสำหรับระบบหน่วยความจำดาวเทียม	81
6.2	ออกแบบวิธีการทดสอบ	83
6.2.1	ประเมินความถูกต้องของการสร้างปริภูมิสถานะ	84
6.2.2	ประเมินประสิทธิภาพของเครื่องสร้างปริภูมิสถานะ	84
6.2.3	ประเมินประสิทธิภาพการทำงานของเครื่องมือการทวนสอบ	84

6.3 ผลการดำเนินการทดสอบ .....	85
6.3.1 ผลทดสอบการประเมินความถูกต้องของการสร้างปริญญีสถานะ .....	85
6.3.2 ผลการทดสอบการประเมินประสิทธิภาพของเครื่องสร้างปริญญีสถานะ .....	86
6.3.3 ผลทดสอบการประเมินประสิทธิภาพการทำงานของเครื่องมือการทวนสอบ .....	89
บทที่ 7      สรุปผลการวิจัยและข้อเสนอแนะ .....	100
7.1 สรุปผลการวิจัย.....	100
7.1.1 อภิปรายผลเพิ่มเติม.....	101
7.2 ข้อจำกัดของงานวิจัย .....	103
7.3 ข้อเสนอแนะและแนวทางในการพัฒนา .....	104
บรรณานุกรม.....	105
ภาคผนวก ก. คำอธิบายยูสเคส.....	110
ภาคผนวก ข. การใช้งานระบบการสร้างปริญญีสถานะและการทวนสอบคุณสมบัติ.....	120
ภาคผนวก ค. กราฟปริญญีสถานะสำหรับกรณีศึกษาที่ 2 3 และ 4 .....	135
ภาคผนวก ง. ผลประเมินประสิทธิภาพการตรวจสอบความถูกต้องของข้อมูลนำเข้าโดยใช้เครื่องมือ การทวนสอบ .....	138
ภาคผนวก จ. อภิธานศัพท์.....	140
ประวัติผู้เขียน.....	149

## สารบัญตาราง

	หน้า
ตารางที่ 4.1 ตัวอย่างการกำหนดสูตรแอลทีแอลทีที่ใช้ในการทวนสอบ.....	51
ตารางที่ 4.2 ตัวอย่างของผลลัพธ์การตรวจสอบข้อมูลนำเข้าที่เป็นสูตรแอลทีแอลแบบเฉพาะ .....	52
ตารางที่ 5.1 รายการแสดงยูสเคสทั้งหมด.....	63
ตารางที่ 6.1 ลักษณะของกราฟปริภูมิสถานะของแต่ละกรณีศึกษา.....	71
ตารางที่ 6.2 รายชื่อเพลสและคำอธิบายของกรณีศึกษาที่ 1 ระบบเครื่องรับจ่ายเงินอัตโนมัติ.....	72
ตารางที่ 6.3 รายชื่อเพลสและคำอธิบายของกรณีศึกษาที่ 2 ระบบการรับประทานอาหารของ นักปราชญ์ 9 คน .....	75
ตารางที่ 6.4 รายชื่อเพลสและคำอธิบายของกรณีศึกษาที่ 3 ระบบสัญญาณไฟจราจรสี่ต้น .....	78
ตารางที่ 6.5 รายชื่อเพลสและคำอธิบายของกรณีศึกษาที่ 4 ระบบหน่วยความจำดาวเทียม .....	82
ตารางที่ 6.6 ผลการประเมินความถูกต้องของการสร้างปริภูมิสถานะโดยใช้เครื่องสร้างปริภูมิสถานะ .....	86
ตารางที่ 6.7 ผลการประเมินประสิทธิภาพของเครื่องสร้างปริภูมิสถานะและแบบจำลองพื้นฐาน .....	87
ตารางที่ 6.8 ผลการประเมินความถูกต้องของการสร้างปริภูมิสถานะโดยใช้เครื่องสร้างปริภูมิสถานะ ของแบบจำลองพื้นฐาน .....	88
ตารางที่ 6.9 ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง สำหรับกรณีศึกษาที่ 1 .....	89
ตารางที่ 6.10 ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง สำหรับกรณีศึกษาที่ 2 .....	91
ตารางที่ 6.11 ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง สำหรับกรณีศึกษาที่ 3 .....	93
ตารางที่ 6.12 ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง สำหรับกรณีศึกษาที่ 4 .....	95
ตารางที่ ก.1 คำอธิบายยูสเคส Browse directory .....	110

ตารางที่ ก.2 คำอธิบายยูสเคส Generate state space .....	111
ตารางที่ ก.3 คำอธิบายยูสเคส Validate user input .....	112
ตารางที่ ก.4 คำอธิบายยูสเคส Show generate state space .....	113
ตารางที่ ก.5 คำอธิบายยูสเคส Export generated state space .....	114
ตารางที่ ก.6 คำอธิบายยูสเคส Verify LTL property .....	115
ตารางที่ ก.7 คำอธิบายยูสเคส Fill LTL formula .....	116
ตารางที่ ก.8 คำอธิบายยูสเคส Import LTL formula.....	116
ตารางที่ ก.9 คำอธิบายยูสเคส Show verification result.....	117
ตารางที่ ก.10 คำอธิบายยูสเคส Delete verification result.....	118
ตารางที่ ก.11 คำอธิบายยูสเคส Show the total number of the generated exaction paths .....	118
ตารางที่ ก.12 คำอธิบายยูสเคส Show example of specific LTL formulas .....	119
ตารางที่ ก.13 คำอธิบายยูสเคส Show status of the system .....	119
ตารางที่ ง.1 ผลประเมินประสิทธิภาพการตรวจสอบข้อมูลนำเข้าโดยใช้เครื่องมือการทวนสอบ .....	138

## สารบัญรูปร่างภาพ

	หน้า
รูปที่ 2.1 ภาพรวมของกระบวนการทวนสอบด้วยโมเดลเชิงกึ่ง [3] .....	8
รูปที่ 2.2 สัญกรณ์ขององค์ประกอบของแบบจำลองเพทรีเน็ต [21] .....	9
รูปที่ 2.3 ตัวอย่างพฤติกรรมของเพทรีเน็ต.....	10
รูปที่ 2.4 ตัวอย่างแบบจำลองเพทรีเน็ตสำหรับระบบไฟจราจรสองต้น.....	11
รูปที่ 2.5 ตัวอย่างกราฟปริภูมิสถานะของระบบไฟจราจรสองต้น.....	12
รูปที่ 2.6 ตัวอย่างกราฟปริภูมิสถานะและต้นไม้เชิงการคำนวณ [25].....	13
รูปที่ 2.7 แบบจำลองเวลาแบบเส้นตรงของประพจน์ที่เขียนด้วยตัวดำเนินการเชิงเวลา [26].....	15
รูปที่ 2.8 ลำดับการนำแบบจำลองเพทรีเน็ตมาสร้างเป็นกราฟปริภูมิสถานะ.....	17
รูปที่ 2.9 โครงสร้างสถาปัตยกรรมของแอลเอสทีเอ็มและสมการของแอลเอสทีเอ็ม [33] .....	21
รูปที่ 2.10 ตัวอย่างจุดตัดของเกณฑ์ที่ดีที่สุดโดยใช้ทฤษฎีของ J-index บนเส้นโค้ง ROC [40] .....	25
รูปที่ 3.1 แผนภาพขั้นตอนการพัฒนาเครื่องสร้างปริภูมิสถานะ .....	31
รูปที่ 3.2 แผนภาพกิจกรรมการเตรียมข้อมูลสำหรับการเรียนรู้ของแบบจำลอง LSTM-MLC.....	32
รูปที่ 3.3 ตัวอย่างแบบจำลองเพทรีเน็ตและการแจกแจงมาร์กกิง .....	33
รูปที่ 3.4 ตัวอย่างการเข้าถึงมาร์กกิงทั้งหมดของแบบจำลองเพทรีเน็ต.....	34
รูปที่ 3.5 ตัวอย่างการถอดรหัสการเข้าถึงมาร์กกิงทั้งหมดเพื่อให้ได้เป็นกราฟปริภูมิสถานะ .....	35
รูปที่ 3.6 ตัวอย่างเส้นทางย่อยที่เป็นไปได้ทั้งหมดที่ได้จากวิธีการหน้าต่างบานเลื่อน .....	36
รูปที่ 3.7 ตัวอย่างของกระบวนการทำโทเคนไนเซชันสำหรับ .....	37
รูปที่ 3.8 ตัวอย่างการตัดเส้นทางย่อยเพื่อให้ได้คุณลักษณะอินพุตและป้ายกำกับเอาต์พุต .....	38
รูปที่ 3.9 ตัวอย่างการเข้ารหัสไบนารีเวกเตอร์สำหรับป้ายกำกับเอาต์พุต .....	39
รูปที่ 3.10 ตัวอย่างชุดข้อมูลสำหรับแบบจำลอง LSTM-MLC.....	40
รูปที่ 3.11 โครงสร้างสถาปัตยกรรมของแบบจำลอง LSTM-MLC .....	42

รูปที่ 3.12 ตัวอย่างการทำนายสถานะถัดไปโดยใช้เครื่องสร้างปฏิภูมิสถานะ	45
รูปที่ 3.13 ตัวอย่างการสร้างปฏิภูมิสถานะโดยเครื่องสร้างปฏิภูมิสถานะ	46
รูปที่ 4.1 แผนภาพขั้นตอนการพัฒนาเครื่องมือการทวนสอบคุณสมบัติแบบเฉพาะ	47
รูปที่ 4.2 ตัวอย่างผลลัพธ์ของฟังก์ชันอีแอลเอฟที่แมปชื่อเพลสไปยังสถานะที่สอดคล้องกัน	50
รูปที่ 4.3 แผนภาพกิจกรรมการทวนสอบคุณสมบัติด้านความปลอดภัย	54
รูปที่ 4.4 แผนภาพกิจกรรมการทวนสอบคุณสมบัติด้านความดำเนินชีวิต	55
รูปที่ 4.5 แผนภาพกิจกรรมการทวนสอบคุณสมบัติด้านความสัมพันธ์แบบที่หนึ่ง	56
รูปที่ 4.6 แผนภาพกิจกรรมการทวนสอบคุณสมบัติด้านความสัมพันธ์แบบที่สอง	57
รูปที่ 4.7 แผนภาพกิจกรรมการทวนสอบสูตรแอลทีแอลประเภท Until	58
รูปที่ 5.1 แผนภาพยูสเคสของระบบการสร้างปฏิภูมิสถานะและการทวนสอบคุณสมบัติ	62
รูปที่ 5.2 แผนภาพกิจกรรมการพัฒนาการสร้างเครื่องสร้างปฏิภูมิสถานะ	64
รูปที่ 5.3 แผนภาพระบบย่อยของระบบการสร้างปฏิภูมิสถานะและการทวนสอบคุณสมบัติของระบบ	64
รูปที่ 5.4 แผนภาพดีพลอยमेंท์ของระบบการสร้างปฏิภูมิสถานะและการทวนสอบคุณสมบัติ	66
รูปที่ 5.5 แผนภาพกิจกรรมการสร้างปฏิภูมิสถานะ	68
รูปที่ 5.6 แผนภาพกิจกรรมการทวนสอบคุณสมบัติการทำงานของระบบ	68
รูปที่ 6.1 แบบจำลองเพทรีเน็ตสำหรับระบบเครื่องรับจ่ายเงินอัตโนมัติ	72
รูปที่ 6.2 การเข้าถึงมาร์กกิงทั้งหมดของระบบเครื่องรับจ่ายเงินอัตโนมัติ	73
รูปที่ 6.3 กราฟปฏิภูมิสถานะของระบบเครื่องรับจ่ายเงินอัตโนมัติ	74
รูปที่ 6.4 แบบจำลองเพทรีเน็ตสำหรับระบบการรับประทานอาหารของนักปราชญ์ 9 คน	75
รูปที่ 6.5 ตัวอย่างกราฟปฏิภูมิสถานะของระบบการรับประทานอาหารของนักปราชญ์ 9 คน	77
รูปที่ 6.6 แบบจำลองเพทรีเน็ตสำหรับระบบสัญญาณไฟจราจรสี่ต้น	78
รูปที่ 6.7 ตัวอย่างกราฟปฏิภูมิสถานะของระบบสัญญาณไฟจราจรสี่ต้น	80
รูปที่ 6.8 แบบจำลองเพทรีเน็ตของระบบหน่วยความจำดาวเทียม	81

รูปที่ 6.9 ตัวอย่างกราฟปริภูมิสถานะของระบบหน่วยความจำดาวเทียม .....	83
รูปที่ 7.1 แบบจำลองเพทรีเน็ตสำหรับกรณีของการสร้างโทเคนแบบอัตโนมัติ .....	102
รูปที่ ข.1 หน้าจอเริ่มต้นของระบบการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติ .....	120
รูปที่ ข.2 หน้าจอสำหรับระบบกำลังสร้างปริภูมิสถานะ .....	121
รูปที่ ข.3 หน้าจอแสดงผลการสร้างปริภูมิสถานะเรียบร้อยและตัวอย่างผลลัพธ์ปริภูมิสถานะ .....	122
รูปที่ ข.4 หน้าจอแสดงผลลัพธ์การสร้างปริภูมิสถานะ .....	122
รูปที่ ข.5 หน้าจอแสดงหน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อผู้ใช้ระบุไอดีเรกทอรีไม่ถูกต้อง .....	123
รูปที่ ข.6 หน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อกำหนดเส้นทางดำเนินงานมีความยาวเกินกว่าที่กำหนด .....	123
รูปที่ ข.7 หน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อผู้ใช้ไม่ระบุเส้นทางดำเนินงานเริ่มต้น .....	124
รูปที่ ข.8 หน้าต่างแจ้งเตือนข้อผิดพลาดผู้ใช้ระบุเส้นทางดำเนินงานเริ่มต้นไม่ถูกต้อง .....	124
รูปที่ ข.9 หน้าต่างแจ้งเตือนข้อผิดพลาดผู้ใช้ระบุจำนวนความยาวไม่ถูกต้อง .....	125
รูปที่ ข.10 หน้าจอแสดงแท็บการทวนสอบคุณสมบัติ .....	125
รูปที่ ข.11 หน้าจอเมื่อระบบดำเนินการทวนสอบเสร็จเรียบร้อยแล้ว .....	126
รูปที่ ข.12 หน้าจอแสดงผลการทวนสอบเป็นที่ไม่พอใจ .....	127
รูปที่ ข.13 ตัวอย่างรูปแบบของเอกสารสูตรแอลทีแอล .....	127
รูปที่ ข.14 หน้าจอแสดงระบบกำลังดำเนินการทวนสอบเอกสารสูตรแอลทีแอล .....	128
รูปที่ ข.15 หน้าจอแสดงผลการทวนสอบเอกสารสูตรแอลทีแอล .....	128
รูปที่ ข.16 หน้าจอแสดงการลบผลการทวนสอบ .....	129
รูปที่ ข.17 หน้าจอแสดงผลการตรวจพบข้อผิดพลาดที่กำหนดไม่มีอยู่ในระบบ .....	130
รูปที่ ข.18 หน้าจอแสดงข้อผิดพลาดเมื่อสูตรแอลทีแอลที่ระบุไม่ถูกต้อง .....	130
รูปที่ ข.19 หน้าจอแสดงข้อผิดพลาดเมื่อผู้ใช้ระบุเครื่องหมาย && และ    พร้อมกัน .....	131
รูปที่ ข.20 หน้าจอแสดงผลข้อผิดพลาดเมื่อสูตรแอลทีแอลที่ระบุมีวงเล็บที่ไม่สมดุล .....	131
รูปที่ ข.21 หน้าจอแสดงผลข้อผิดพลาดเมื่อสูตรแอลทีแอลไม่สอดคล้องกับคุณสมบัติที่เลือก .....	132

รูปที่ ข.22 หน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อไม่พบเครื่องสร้างปฏิภูมิสถานะจากไทรเรททอรีที่ระบุ 132

รูปที่ ข.23 หน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อผู้ใช้ไม่เลือกคุณสมบัติที่ต้องการทวนสอบ..... 133

รูปที่ ข.24 หน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อผู้ใช้ไม่ระบุสูตรแอลทีแอลที่ต้องการทวนสอบ ..... 133

รูปที่ ข.25 หน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อรูปแบบของเอกสารสูตรแอลทีแอลไม่ถูกต้อง ..... 134

รูปที่ ค.1 การนำเสนอแผนภาพข้อมูลของกราฟปฏิภูมิสถานะของกรณีศึกษาที่ 2 ระบบการ  
รับประทานอาหารของนักปราชญ์ 9 คน..... 135

รูปที่ ค.2 การนำเสนอแผนภาพข้อมูลของกราฟปฏิภูมิสถานะของกรณีศึกษาที่ 3 ระบบสัญญาณไฟ  
จราจรสี่ต้น..... 136

รูปที่ ค.3 การนำเสนอแผนภาพข้อมูลของกราฟปฏิภูมิสถานะของกรณีศึกษาที่ 4 ระบบหน่วยความจำ  
ดาวเทียม..... 137





## บทที่ 1

### บทนำ

#### 1.1 ที่มาและความสำคัญ

ในปัจจุบันระบบซอฟต์แวร์เป็นเครื่องมือสำคัญที่เข้ามามีบทบาทช่วยเพิ่มความสะดวกสบายให้กับมนุษย์มากขึ้น ส่งผลให้อุตสาหกรรมซอฟต์แวร์ได้ออกแบบและพัฒนาระบบซอฟต์แวร์ออกมาหลากหลายรูปแบบ อีกทั้งยังเพิ่มประสิทธิภาพในการทำงานเพื่อตอบสนองตามความต้องการของผู้ใช้ ซึ่งขนาดของซอฟต์แวร์ที่มีขนาดเล็กหรือขนาดใหญ่ก็เกิดความซับซ้อนขึ้นได้ โดยเฉพาะระบบที่มีขนาดใหญ่มักจะมี ความซับซ้อนสูง ทำให้มีแนวโน้มที่จะเกิดข้อผิดพลาดได้ง่าย ข้อผิดพลาดที่เกิดขึ้นหลังกระบวนการพัฒนาระบบซอฟต์แวร์จะส่งผลต่ออัตราสูญเสียทางด้านเวลา ต้นทุนหรือทรัพยากรมากกว่าการพบข้อผิดพลาดก่อนลงมือพัฒนาระบบซอฟต์แวร์ ด้วยเหตุนี้ อุตสาหกรรมซอฟต์แวร์จึงเริ่มเห็นความสำคัญของการทวนสอบการทำงานของระบบก่อนที่จะเข้าสู่กระบวนการลงมือพัฒนาระบบเพื่อลดข้อผิดพลาดที่อาจเกิดขึ้น โดยปกติ การทวนสอบสามารถแบ่งออกเป็นสองแบบ [1] คือ แบบแรก การทวนสอบอย่างไม่เป็นทางการ (Informal verification) เป็นการทวนสอบโดยให้ผู้เชี่ยวชาญเป็นผู้ทวนสอบระบบ เช่น ผู้ทดสอบ (Tester) หรือ ผู้ใช้งาน (User) เป็นต้น ซึ่งอาจทำให้มี การทำงานบางส่วนถูกละเลยการทวนสอบ เนื่องจากอาจเกิดจากความผิดพลาดของมนุษย์ (Human error) ขึ้นมาได้ ในขณะที่แบบที่สอง การทวนสอบเชิงรูปนัย (Formal verification) เป็นการทวนสอบโดยใช้เครื่องมือการพิสูจน์ทางคณิตศาสตร์มาช่วยทวนสอบพฤติกรรมการทำงานของระบบ ซึ่งสามารถทวนสอบทุกการทำงานที่เป็นไปได้ทั้งหมดของระบบ โดยใช้วิธีการเชิงรูปนัย (Formal method) ดังนั้น กระบวนการทวนสอบเชิงรูปนัย (Formal verification process) จึงเป็น กระบวนการที่สำคัญของการตรวจหาข้อผิดพลาดหรือทวนสอบหาคุณสมบัติการทำงานทั้งหมดของระบบที่สนใจ ก่อนเริ่มลงมือพัฒนาระบบ ตัวอย่างคุณสมบัติที่ใช้ในการทวนสอบระบบ เช่น ความปลอดภัย (Safety) ความดำเนินชีวิต (Liveness) และความสัมพันธ์ (Correlation) เป็นต้น นอกจากนี้ คุณสมบัติด้านความปลอดภัยและความดำเนินชีวิตเป็นคุณสมบัติพื้นฐานของคุณสมบัติ อื่นๆ และมักใช้การตรวจสอบความถูกต้องโดยทั่วไปของระบบซอฟต์แวร์ [2] หากระบบซอฟต์แวร์มีการทวนสอบพฤติกรรมการทำงานก่อนลงมือพัฒนาจะสามารถช่วยลดความพยายามที่จะต้องสูญเสียไป โดยเฉพาะอย่างยิ่งสำหรับระบบซอฟต์แวร์ขนาดใหญ่และซับซ้อน

โมเดลเช็กกิง (Model checking) [3] เป็นวิธีการเชิงรูปนัยที่ได้รับความนิยมสำหรับการทวนสอบพฤติกรรมการทำงานของระบบที่สนใจว่ามีการทำงานที่เป็นไปตามคุณสมบัติเป้าหมายที่ต้องการหรือไม่ การทวนสอบโดยใช้โมเดลเช็กกิง ประกอบด้วย แบบจำลองเชิงรูปนัยของระบบ (Formal system model) และคุณสมบัติการทำงานของระบบที่ต้องการตรวจสอบซึ่งสามารถเขียนสูตร

ตรรกศาสตร์เชิงเวลาแบบลำดับหรือ แอลทีแอล (Linear temporal logic หรือ LTL) โดยหลักการทำงานของโมเดลเช็กกิงใช้วิธีพิสูจน์โดยการแจกแจงปริภูมิสถานะ (State space) ของพฤติกรรมการทำงานของระบบที่สนใจทั้งหมด โดยปริภูมิสถานะที่ถูกแจกแจงออกมาสามารถเขียนให้อยู่ในรูปแบบของกราฟปริภูมิสถานะ เพื่อแสดงให้เห็นถึงสถานะและลำดับการเปลี่ยนแปลงสถานะที่ถูกแปลงมาจากแบบจำลองเชิงรูปนัยที่กำหนด จากนั้นนำรูปแบบของคุณสมบัติของการทำงานที่ต้องการทวนสอบไปค้นอย่างละเอียดถี่ถ้วนในปริภูมิสถานะและตอบกลับว่ายอมรับคุณสมบัติเป้าหมายนั้นหรือไม่ ซึ่งผู้ทวนสอบไม่จำเป็นต้องมีความรู้ทางด้านตรรกศาสตร์มากนัก อย่างไรก็ตามระบบที่มีอยู่จริงมักมีขนาดใหญ่และมีความซับซ้อน ซึ่งเป็นสาเหตุหลักของการเติบโตแบบทวีคูณของสถานะและการเปลี่ยนสถานะในปริภูมิสถานะที่มีมากมายมหาศาลและใช้หน่วยความจำเป็นจำนวนมาก ส่งผลให้โมเดลเช็กกิงไม่ได้รับการจัดสรรทรัพยากรที่เพียงพอ ซึ่งในที่สุดโมเดลเช็กกิงก็หยุดการทำงานและไม่สามารถทวนสอบระบบได้ ข้อจำกัดนี้เรียกว่า การระเบิดของปริภูมิสถานะ (State space explosion) ดังนั้น โมเดลเช็กกิงที่มีอยู่มักจะใช้เทคนิคต่างๆ เพื่อจัดการกับการระเบิดของปริภูมิสถานะ เช่น การลดจำนวนปริภูมิสถานะ การตัดแต่งปริภูมิสถานะและเทคนิคอื่นๆ [4] เป็นต้น

ในปีที่ผ่านมาโครงข่ายประสาทเทียมแบบวนซ้ำ (Recurrent neural network) ประเภทโครงข่ายความจำระยะสั้นแบบยาวหรือแอลเอสทีเอ็ม (Long short-term memory network หรือ LSTM) [5] ถูกพัฒนาให้มีความสามารถในการเรียนรู้การขึ้นต่อกันของลำดับและรู้จำลำดับข้อมูลได้อย่างมีประสิทธิภาพ เพื่อจัดการปัญหาการทำนายอย่างมีลำดับสำหรับข้อมูลอนุกรมเวลา (Time series data) มีหลายงานวิจัย [6-9] นำเสนอแบบจำลองการทำนายลำดับต่อลำดับ (Sequence to sequence) โดยใช้แอลเอสทีเอ็มเพื่อเรียนรู้ลำดับของข้อมูลอนุกรมและทำนายลำดับของคำหรือป้าย นอกจากนี้มีหลายงานวิจัยที่ได้นำแอลเอสทีเอ็มมาใช้ทำนายการจำแนกแบบหลายป้าย (Multi-label classification) เช่น การจัดหมวดหมู่ข้อความ (Text categorization) [10, 11] และในการวินิจฉัยโรคทางการแพทย์ (Diagnosis disease for medical) [12] อย่างไรก็ตาม ข้อมูลที่ใช้จริงมักมีลักษณะที่เป็นข้อมูลไม่สมดุล (Imbalance data) เช่น สอนให้แอลเอสทีเอ็มเรียนรู้ผลไม้ที่เป็น ส้ม มากกว่าองุ่น เมื่อให้ทำนายผลไม้อยู่ในรูปแบบ ปรากฏว่าทำนายได้เฉพาะ ส้ม ถึงแม้ว่าในรูปนั้นมีองุ่นด้วยเช่นกัน ส่งผลให้การทำนายการจำแนกแบบหลายป้ายให้ถูกต้องและครบถ้วนไม่ใช่เรื่องง่าย ดังนั้นจึงเป็นความท้าทายที่จะทำนายผลลัพธ์ได้ถูกต้อง หนึ่งในวิธีการที่ใช้กันอยู่แพร่หลาย คือ กลยุทธ์การหาขีดแบ่ง (Threshold strategy) ซึ่งเป็นกระบวนการหาค่าคงที่เพื่อใช้เป็นขีดแบ่งสำหรับแบ่งว่าผลลัพธ์ของหมวดหมู่หรือป้ายใดเป็นผลลัพธ์การทำนายที่แท้จริง เนื่องจาก แบบจำลองแมชชีนเลิร์นนิงจะส่งคืนค่าผลลัพธ์การทำนายของแต่ละป้ายที่เป็นอยู่ในรูปแบบของตัวเลข เช่น ค่าความน่าจะเป็น ซึ่งยังไม่ได้เป็นผลการทำนายที่ชัดเจน ดังนั้น ขีดแบ่งที่ถูกกำหนดจะเป็นค่าคงที่ เพื่อแบ่งว่าป้ายใดควรได้รับการจำแนกให้เป็นผลลัพธ์การทำนายที่แท้จริง โดยปกติ มีการกำหนดขีดแบ่งหลายแบบ

เช่น การใช้ค่าโดยปริยาย (Default value) ที่มีค่าเท่ากับ 0.5 [13, 14] การใช้ค่าที่มากที่สุดของผลลัพธ์การทำนายของทุกป้าย [15] และการจัดลำดับคะแนนของผลลัพธ์การทำนายของทุกป้าย [16] เป็นต้น อย่างไรก็ตาม การใช้ค่าดังกล่าวอาจมีประสิทธิภาพการทำนายที่ไม่ค่อยดีเท่าไรนัก เพราะต้องพิจารณาผลลัพธ์การทำนายของทุกป้ายเพื่อให้ได้ขีดแบ่งที่เหมาะสมกับลักษณะของข้อมูล เช่น การนำผลลัพธ์การทำนายมาคำนวณหาพื้นที่ใต้กราฟเพื่อเลือกจุดตัดที่ดีที่สุดให้เป็นขีดแบ่ง เป็นต้น

จากที่กล่าวไว้ข้างต้น การทวนสอบโดยใช้โมเดลเชิงกึ่งที่มีอยู่เป็นวิธีการแจกแจงปริภูมิสถานะแบบเก่าโดยมีการทำงานตามอัลกอริทึมที่นักพัฒนาเขียนขึ้นมา อีกทั้งระบบที่มีขนาดใหญ่และมีความซับซ้อน โดยระบบที่มีความซับซ้อนสามารถสังเกตได้จากจำนวนของสถานะที่เกิดขึ้นภายในระบบและลำดับการเปลี่ยนแปลงของสถานะที่มีเป็นจำนวนมากและเมื่อนำมาเขียนเป็นกราฟปริภูมิสถานะก็จะพบเส้นทางโยงใยแทรกสลับ ทำให้ตามรอยได้ยากและอาจส่งผลให้เกิดการระเบิดของปริภูมิสถานะในโมเดลเชิงกึ่ง ดังนั้น งานวิจัยนี้จึงได้ผสมผสานแนวคิดของแอลเอสทีเอ็มและการจำแนกแบบหลายป้ายสำหรับพัฒนาเป็นเครื่องสร้างปริภูมิสถานะ (State space generator) สำหรับการทำโมเดลเชิงกึ่งแบบแอลทีแอล เพื่อบรรเทาปัญหาการระเบิดของปริภูมิสถานะในโมเดลเชิงกึ่งแบบดั้งเดิม โดยเริ่มจากการออกแบบและพัฒนาแบบจำลองแอลเอสทีเอ็มให้มีความสามารถในการจำแนกสถานะแบบหลายป้าย เพื่อให้เหมาะสมกับการสร้างปริภูมิสถานะของระบบจริงแบบตามที่ต้องการ (On-demand) จากนั้น ออกแบบและพัฒนาเครื่องมือการทวนสอบคุณสมบัติการทำงานของระบบสำหรับการทำโมเดลเชิงกึ่งแบบแอลทีแอล โดยนำเครื่องสร้างปริภูมิสถานะมาใช้เทคนิคแบบออน-เดอะ-ฟลาย (On-the-fly) สำหรับสร้างปริภูมิสถานะเพื่อใช้สำหรับทวนสอบคุณสมบัติพื้นฐานของระบบซึ่งถูกกำหนดโดยการเขียนด้วยสูตรแอลทีแอลแบบเฉพาะ ได้แก่ ความปลอดภัย (Safety) ความดำเนินชีวิต (Liveness) และความสัมพันธ์ (Correlation) และเพื่อให้ครอบคลุมตัวดำเนินการเชิงเวลาสูตรแอลทีแอลประเภทจนกระทั่ง (Until) ถูกกำหนดให้สามารถทวนสอบได้เช่นกัน โดยการทวนสอบใช้เทคนิคออน-เดอะ-ฟลาย เป็นการค้นรูปแบบคุณสมบัติที่ต้องการทวนสอบ โดยค้นหาที่สถานะแบบมองไปข้างหน้า (Look ahead) โดยเริ่มต้นจากสถานะเริ่มต้นจนกระทั่งพบสถานะเป้าหมาย ซึ่งทำให้ใช้หน่วยความจำน้อยลง ทำให้ได้เครื่องสร้างปริภูมิสถานะและเครื่องมือการทวนสอบคุณสมบัติ ด้วยเหตุนี้ เพื่อรวมความสามารถของการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติของระบบ ผู้วิจัยได้สร้างเป็นระบบการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติของระบบโดยการเรียกใช้เครื่องสร้างปริภูมิสถานะที่ถูกพัฒนาขึ้นมา

## 1.2 วัตถุประสงค์

- 1) เพื่อออกแบบแบบจำลองแอลเอสทีเอ็มที่สามารถจำแนกสถานะแบบหลายป้าย

- 2) เพื่อพัฒนาเครื่องสร้างปริภูมิสถานะของการทำงานของระบบที่สนใจ โดยใช้วิธีการผสมผสานแอลเอสทีเอ็มที่สามารถจัดการกับการจำแนกหลายป้าย
- 3) เพื่อพัฒนาเครื่องมือการทวนสอบคุณสมบัติพื้นฐานของระบบที่สนใจ โดยใช้เทคนิคออน-เดอะ-ฟลาย สำหรับทวนสอบคุณสมบัติที่ต้องการ ซึ่งจะกำหนดโดยการเขียนเป็นสูตรแอลทีแอล

### 1.3 ขอบเขตการดำเนินงาน

- 1) แบบจำลองเชิงรูปนัยต้นฉบับเขียนด้วยเพทรีเน็ตแบบธรรมดาสำหรับจำลองระบบแบบจำกัด (Finite system) บนเครื่องมือที่นำ
- 2) เครื่องสร้างปริภูมิสถานะถูกสร้างจากโดยการผสมผสานแอลเอสทีเอ็มที่สามารถทำนายการจำแนกหลายป้ายและมีการหาขีดแบ่งที่เหมาะสม
- 3) ปริภูมิสถานะที่สร้างขึ้นจะแสดงผลในรูปแบบของชุดของเส้นทางดำเนินงานของระบบ
- 4) สูตรแอลทีแอลสำหรับใช้ในการทวนสอบ มีคุณสมบัติดังนี้
  - 4.1) ตัวเชื่อมประพจน์ ประกอบด้วย นิเสธ (Negation) และ (Conjunction) หรือ (Disjunction) และการให้เหตุผล (Implication)
  - 4.2) ตัวดำเนินการเชิงเวลา ประกอบด้วย เสมอ (Always) ถัดไป (Next) ในที่สุด (Finally) และจนกระทั่ง (Until)
  - 4.3) ประพจน์เดี่ยวจะระบุโดยใช้เพลสเท่านั้น
- 5) เครื่องมือการทวนสอบคุณสมบัติของการทำงานของระบบ จะมีคุณสมบัติดังนี้
  - 5.1) รับข้อมูลเป็นปริภูมิสถานะที่สร้างขึ้นจากเครื่องสร้างปริภูมิสถานะและคุณสมบัติของระบบที่เขียนด้วยสูตรแอลทีแอลแบบเฉพาะ
  - 5.2) คุณสมบัติพื้นฐานที่ใช้ทวนสอบระบบ ประกอบด้วย คุณสมบัติทางด้านความปลอดภัย (Safety) ความดำเนินชีวิต (Liveness) และความสัมพันธ์ (Correlation)
- 6) เครื่องสร้างปริภูมิสถานะสนับสนุนการสร้างปริภูมิสำหรับระบบที่มีสถานะไม่น้อยกว่า 5,000 สถานะ
- 7) เครื่องมือการทวนสอบคุณสมบัติการทำงานของระบบโดยใช้วิธีการออน-เดอะ-ฟลาย (On-the-fly)
- 8) เครื่องมือที่ถูกพัฒนาขึ้นมาถูกนำไปเปรียบเทียบกับประสิทธิภาพกับโมเดลเช็กกิงที่มีอยู่ เช่น เครื่องมือที่นำ เป็นต้น

### 1.4 ขั้นตอนการดำเนินงาน

- 1) ศึกษาการใช้งานเครื่องมือที่นำ

- 2) ค้นคว้าโครงข่ายประสาทเทียมที่เหมาะสมกับลักษณะของข้อมูลที่ใช้ในงานวิจัย
- 3) ศึกษาการจัดเตรียมข้อมูลที่ใช้สำหรับการเรียนรู้
- 4) ศึกษาวิธีการทำงานของแอลเอสทีเอ็ม
- 5) ศึกษาวิธีการจำแนกแบบหลายป้ายและการวัดประสิทธิภาพ
- 6) ค้นคว้าวิธีการหาขีดแบ่งที่เหมาะสมและนำมาประยุกต์ใช้สำหรับการจำแนกแบบหลายป้าย
- 7) ออกแบบแบบจำลองแอลเอสทีเอ็มที่สามารถจำแนกแบบหลายป้าย
- 8) พัฒนาเครื่องสร้างปริภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอลทีแอลโดยใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้าย
- 9) พัฒนาเครื่องมือการทวนสอบที่ใช้ความสามารถของเครื่องสร้างปริภูมิสถานะ
- 10) ออกแบบวิธีการดำเนินการทดสอบและดำเนินการทดสอบ
- 11) วิเคราะห์และสรุปผลการทดลอง

#### 1.5 ประโยชน์ที่ได้รับ

- 1) ได้กระบวนการหาขีดแบ่งที่เหมาะสมแบบใหม่ ที่สามารถนำมาใช้สำหรับการจำแนกสถานะแบบหลายป้าย
- 2) ได้วิธีการออกแบบและพัฒนาแบบจำลองแอลเอสทีเอ็มที่สามารถจำแนกแบบหลายป้าย
- 3) ได้เครื่องสร้างปริภูมิสถานะที่ถูกพัฒนาจากการผสมผสานแบบจำลองแอลเอสทีเอ็มที่สามารถจำแนกแบบหลายป้าย
- 4) ได้เครื่องมือสำหรับการทวนสอบที่สามารถทวนสอบการคุณสมบัติพื้นฐานของระบบโดยสูตรแอลทีแอลที่กำหนด
- 5) ได้เครื่องสร้างปริภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอลทีแอลโดยใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้ายที่สามารถช่วยในการบรรเทา การระเบิดของปริภูมิสถานะในโมเดลเชิงกึ่งได้

#### 1.6 การจัดเรียงของวิทยานิพนธ์

เนื้อหาของวิทยานิพนธ์ฉบับนี้แบ่งออกเป็น 7 บท ดังต่อไปนี้

- บทที่ 1 อธิบายที่มาและความสำคัญของงานวิจัย วัตถุประสงค์ ขอบเขต ขั้นตอนการดำเนินงานและประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย
- บทที่ 2 อธิบายเกี่ยวกับความรู้ที่เกี่ยวข้องสำหรับการทวนสอบเชิงรูปนัย หลักการของแบบจำลองแอลเอสทีเอ็มและการจำแนกแบบหลายป้ายและส่วนสุดท้ายของบทได้กล่าวถึงงานวิจัยที่เกี่ยวข้อง

บทที่ 3 อธิบายแนวคิดการพัฒนาเครื่องสร้างปริญญีสถานะสำหรับโมเดลเชิงกึ่งแบบแอลทีแอลโดยใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้าย

บทที่ 4 อธิบายแนวคิดการพัฒนาเครื่องมือการทวนสอบคุณสมบัติโดยใช้เครื่องสร้างปริญญีสถานะ

บทที่ 5 อธิบายวิธีการออกแบบสถาปัตยกรรมและการพัฒนาระบบการสร้างปริญญีสถานะและการทวนสอบคุณสมบัติ

บทที่ 6 อธิบายวิธีการออกแบบกรณีศึกษา วิธีการทดสอบและการดำเนินการทดลอง

บทที่ 7 สรุปผลการทดลอง อภิปรายเกี่ยวกับข้อจำกัดและข้อเสนอแนะ



## บทที่ 2

### ทฤษฎีที่เกี่ยวข้องและงานวิจัยที่เกี่ยวข้อง

ในบทนี้ นำทฤษฎีและงานวิจัยที่เกี่ยวข้อง โดยหัวข้อที่ 2.1 นำเสนอทฤษฎีพื้นฐานที่ได้นำมาประยุกต์ใช้กับงานวิจัยนี้ แบ่งออกเป็น ทฤษฎีที่เกี่ยวข้องกับการทวนสอบเชิงรูปนัยและทฤษฎีที่เกี่ยวข้องกับแบบจำลองแอลเอสทีเอ็มและการจำแนกแบบหลายและในหัวข้อที่ 2.2 นำเสนองานวิจัยที่เกี่ยวข้องกับงานวิจัยนี้

#### 2.1 ทฤษฎีที่เกี่ยวข้อง

ผู้วิจัยได้ศึกษาค้นคว้าเกี่ยวกับทฤษฎีที่เกี่ยวข้องกับงานวิจัยนี้ โดยสามารถแบ่งออกเป็น 2 ทฤษฎีหลัก ได้แก่ ทฤษฎีที่เกี่ยวข้องกับการทวนสอบเชิงรูปนัยและทฤษฎีที่เกี่ยวข้องเกี่ยวกับแบบจำลองแอลเอสทีเอ็มและการจำแนกแบบหลายป้าย ซึ่งได้อธิบายรายละเอียดในลำดับถัดไป

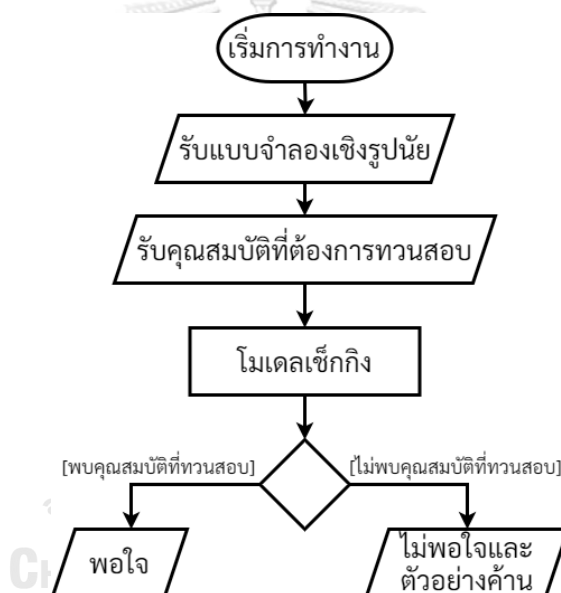
##### 2.1.1 ทฤษฎีที่เกี่ยวข้องกับการทวนสอบเชิงรูปนัย

ในส่วนนี้ นำเสนอทฤษฎีพื้นฐานของการทวนสอบเชิงรูปนัย ซึ่งจะประกอบด้วย โมเดลเช็กกิง เพทรีเน็ต ปริภูมิสถานะและตรรกศาสตร์เชิงเวลาแบบลำดับหรือแอลทีแอล

##### 2.1.1.1 โมเดลเช็กกิง (Model checking)

โมเดลเช็กกิง (Model checking) [3] เป็นวิธีการเชิงรูปนัย (Formal method) ที่ถูกใช้งานได้จริงมากกว่าวิธีการแบบเป็นทางการอื่นๆ เช่น การพิสูจน์ทฤษฎีบท (Theorem proving) และการตรวจสอบความสมมูล (Equivalence checking) เป็นต้น อย่างไรก็ตาม การทวนสอบโดยใช้โมเดลเช็กกิงใช้วิธีการพิสูจน์โดยวิเคราะห์ปริภูมิสถานะ ทำให้ผู้ใช้ไม่จำเป็นต้องมีความรู้ในการพิสูจน์ทางคณิตศาสตร์มากนัก การทวนสอบโดยใช้โมเดลเช็กกิงจำเป็นต้องกำหนดแบบจำลองเชิงรูปนัยที่จำลองการทำงานของระบบที่สนใจและคุณสมบัติที่ต้องการทวนสอบที่สามารถเขียนโดยใช้สูตรตรรกศาสตร์เชิงเวลาเชิงเส้น ซึ่งสามารถเขียนเป็นสูตรแอลทีแอลหรือ สูตรตรรกศาสตร์ต้นไม้เชิงการคำนวณหรือซีทีแอล (Computational tree logic หรือ CTL) กระบวนการทวนสอบเริ่มจากการนำแบบจำลองเชิงรูปนัยมาแปลงเป็นกราฟสถานะเพื่อทำการแจกแจงปริภูมิสถานะทั้งหมดของระบบ จากนั้นนำรูปแบบของคุณสมบัติที่ต้องการทวนสอบมาค้นหาในปริภูมิสถานะที่แจกแจงออกมาและรายงานผลลัพธ์การทวนสอบระบบ หากพบคุณสมบัติที่ต้องการทวนสอบ โมเดลเช็กกิงรายงานผลว่าพอใจ (Satisfied) ในทางตรงกันข้าม โมเดลเช็กกิงจะรายงานผลว่าไม่พอใจ (Not satisfied) พร้อมแสดงตัวอย่างค้าน (Counterexample) ซึ่งภาพรวมของกระบวนการทวนสอบด้วยโมเดลเช็กกิง แสดงในรูปที่ 2.1 ซึ่งโดยทั่วไปการทวนสอบในโมเดลเช็กกิงที่มีอยู่สามารถทำได้สองแบบ แบบแรก คือ การจำลองทีละ

ขั้นตอน (Simulation) เป็นการนำแบบจำลองเชิงรูปนัยที่กำหนดมาแสดงให้เห็นลำดับการทำงานที่ละเอียดขั้นตอน ทำให้สามารถคาดการณ์ได้ว่าการทำงานส่วนใดเกิดข้อผิดพลาดและได้ผลลัพธ์อย่างรวดเร็ว และใช้จำนวนสถานะไม่มาก แต่การแสดงผลการทำงานอาจจะแสดงได้ไม่ละเอียด ทำให้ไม่พบข้อผิดพลาดและอาจต้องจำลองหลายครั้ง แบบที่สอง คือ การทวนสอบ (Verification) เป็นวิธีการอัตโนมัติที่รูปแบบของคุณสมบัติของระบบที่ต้องการทวนสอบมาวิเคราะห์ในปริภูมิสถานะแบบหมดจด (Exhaustive) ทำให้สามารถทวนสอบคุณสมบัติที่ต้องการได้อย่างละเอียด แต่หากระบบที่สนใจมีขนาดใหญ่และมีความซับซ้อนก็อาจทำให้เกิดการระเบิดของปริภูมิสถานะและไม่สามารถทวนสอบระบบได้ ตัวอย่างเครื่องมือที่สามารถทำโมเดลเช็กกิง เช่น เครื่องมือสปิน (Spin tool) [17] รองรับภาษาเชิงรูปนัยโพลล่า (Promela) [18] เครื่องมือทีน่า (TINA tool) [19] และเครื่องมือซีพีเอ็น (CPN tool) [20] รองรับแบบจำลองเชิงรูปนัยเพทรีเน็ต เป็นต้น



รูปที่ 2.1 ภาพรวมของกระบวนการทวนสอบด้วยโมเดลเช็กกิง [3]

เพทรีเน็ตเป็นแบบจำลองเชิงรูปนัยที่ใช้สัญลักษณ์กราฟิกเพื่อจำลองพฤติกรรมของระบบได้หลากหลายสถานการณ์และช่วยให้ผู้วิเคราะห์ระบบสามารถเข้าใจการทำงานของระบบได้ง่าย อีกทั้งยังสามารถทวนสอบคุณสมบัติของระบบที่สนใจได้โดยใช้โมเดลเช็กกิงและสามารถกำหนดสูตรแอลที่แอลที่แสดงคุณสมบัติของพฤติกรรมการทำงานของระบบที่สนใจและนำรูปแบบของคุณสมบัตินั้นไปค้นหาในปริภูมิสถานะและตอบกลับว่าระบบมีพฤติกรรมที่ยอมรับได้หรือไม่



### 2.1.1.2 เพทรีเน็ต (Petri net)

เพทรีเน็ต (Petri net) ถูกนำเสนอโดยคาร์ล ลอดัม เพทรี (Carl Adam Petri) [21] โดยเป็นเพทรีเน็ตแบบธรรมดา (Ordinary Petri net) ซึ่งไม่มีเรื่องเวลาเข้ามาเกี่ยวข้อง เพทรีเน็ตเป็นแบบจำลองเชิงรูปนัยที่ใช้สัญลักษณ์กราฟิกจำลองพฤติกรรมของระบบได้หลากหลายแบบ เช่น ระบบกระจาย (Distributed system) ระบบพร้อมกัน (Concurrent system) และระบบเรียลไทม์ (Real time system) เป็นต้น นอกจากนี้ การจำลองเพทรีเน็ตยังสามารถใช้สัญลักษณ์กราฟิกเพื่อเป็นตัวแทนการทำงานของระบบ ทำให้ผู้ใช้สามารถเข้าใจและสื่อสารกันได้ง่าย ประกอบด้วย เพลส (Place) เป็นตัวแทนที่บ่งบอกถึงเหตุการณ์หรือสถานการณ์ที่เกิดขึ้นภายในระบบ ณ ขณะนั้น ทรานซิชัน (Transition) เป็นตัวแทนของเงื่อนไขการเกิดเหตุการณ์ที่เป็นจริง โดยมีอินพุตเพลส (Input place) ที่มีเส้นเชื่อม (Arcs) ระหว่างเพลสไปยังทรานซิชัน และเอาต์พุตเพลส (Output place) ที่มีเส้นเชื่อมระหว่างทรานซิชันไปยังเพลสเท่านั้น โดยไม่อนุญาตให้มีเส้นเชื่อมระหว่างเพลสไปยังเพลสหรือจากทรานซิชันไปยังทรานซิชัน นอกจากนี้ บนเส้นเชื่อมสามารถระบุค่าน้ำหนัก (Weight) ของอินพุตเพลสและเอาต์พุตเพลสได้ หากไม่กำหนดจะใช้ค่าโดยปริยายที่มีค่าเท่ากับ 1 และสุดท้ายลูกวงกลมสีดำเรียกว่า โทเคน (Token) ซึ่งการกำหนดโทเคนในเพลสแสดงให้เห็นถึงสถานการณ์หรือเหตุการณ์ที่เกิดขึ้นภายในระบบและเหตุการณ์เริ่มต้นจะมีการกำหนดโทเคนไปในเพลสเริ่มต้นของระบบเรียกว่า มาร์กกิงเริ่มต้น (Initial marking) ซึ่งสัญลักษณ์ขององค์ประกอบของแบบจำลองเพทรีเน็ตแสดงในรูปที่ 2.2



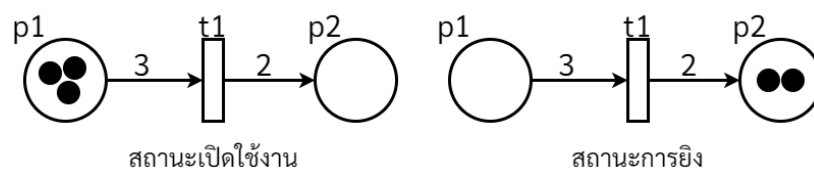
รูปที่ 2.2 สัญลักษณ์ขององค์ประกอบของแบบจำลองเพทรีเน็ต [21]

นิยามอย่างเป็นทางการขององค์ประกอบเชิงสัญลักษณ์ของแบบจำลองเพทรีเน็ตแบบธรรมดา [22] มีทั้งหมด 5 องค์ประกอบ (5-tuple) คือ  $PN = (P, T, F, V, M_0)$  ซึ่งอธิบายโดยสัญลักษณ์ทางคณิตศาสตร์ดังนี้

- $P = \{p_1, p_2, \dots, p_n\}$  คือ ชุดของจำนวนจำกัดของเพลส
- $T = \{t_1, t_2, \dots, t_n\}$  คือ ชุดของจำนวนจำกัดของทรานซิชัน
- $F \subseteq (P \times T) \cup (T \times P)$  ชุดของเส้นทาง (Flow relation หรือ Arcs) คือ ชุดของเส้นทางก่อนเกิดเหตุการณ์จากเพลสไปยังทรานซิชันและชุดของเส้นทางหลังเกิดเหตุการณ์จากทรานซิชันไปยังเพลส
- $V: F \rightarrow \mathbb{N}^+$  คือ น้ำหนักของเส้นเชื่อมที่เป็นจำนวนธรรมชาติ

-  $M_0$  คือ มาร์กกิงเริ่มต้น ซึ่งระบุจำนวนโทเคนที่อยู่ในเพลสในสถานะเริ่มต้น

พฤติกรรมของเพทรีเน็ตมีอยู่สองสถานะ ได้แก่ สถานะแรก คือ สถานะเปิดใช้งาน (Enable state) ซึ่งเป็นเหตุการณ์ก่อนเกิดการยิง โดยจะเกิดขึ้นเมื่ออินพุตเพลสมีจำนวนโทเคนเพียงพอตามค่าน้ำหนักที่อินพุตเพลสต้องการ สถานะที่สอง คือ สถานะการยิง (Firing state) โดยจะเกิดขึ้นเมื่อทรานซิชันอยู่ในสถานะเปิดใช้งานแล้ว จากนั้น ทรานซิชันเข้าสู่สถานะการยิงเพื่อทำการบริโภค (Consume) โทเคนจากอินพุตเพลสและหลังจากนั้นจะผลิต (Produce) โทเคนไปยังเอาต์พุตเพลส โดยผลิตโทเคนที่มีจำนวนเท่ากับค่าน้ำหนักที่เอาต์พุตเพลสต้องการ โดยแสดงในรูปที่ 2.3



รูปที่ 2.3 ตัวอย่างพฤติกรรมของเพทรีเน็ต

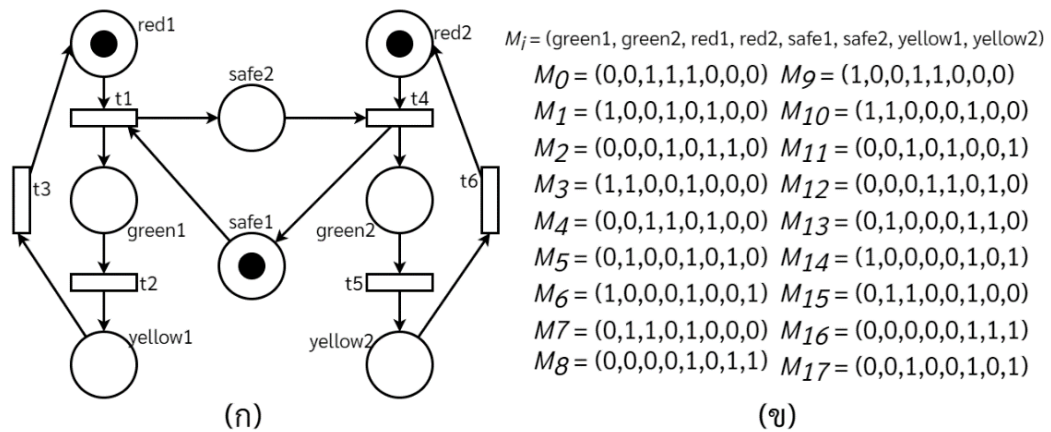
เพลสที่มีโทเคนบรรจุอยู่ภายใน สามารถแสดงพฤติกรรมของระบบ ณ เวลาช่วงหนึ่ง โดยสามารถดูได้จากมาร์กกิง ซึ่งแต่ละมาร์กกิงสามารถจะแสดงถึงสถานะ (State) ของพฤติกรรมของระบบที่จำลองโดยแบบจำลองเพทรีเน็ต ซึ่งสถานะเริ่มต้นของระบบแสดงด้วยมาร์กกิง  $M_0$  ซึ่งสามารถแสดงด้วยสถานะเริ่มต้น  $s_0$  เช่นกัน เพื่อให้ผู้อ่านสามารถทำความเข้าใจได้ง่ายขึ้น ในวิทยานิพนธ์จึงได้ออกแบบและแยกตัวอย่างระบบไฟจราจรสองต้นเป็นกรณีศึกษา โดยการทำงานของแต่ละเสาไฟจราจรเริ่มจากไฟแดง ไฟเขียวและไฟเหลือง ตามลำดับ โดยที่เสาไฟจราจรจะเปลี่ยนเป็นสีเขียวได้ก็ต่อเมื่อ เสาไฟต้นนั้นเป็นสีแดงและมีสถานะที่ปลอดภัย ซึ่งแสดงสัญญาณไฟต่างๆ และความปลอดภัยโดยใช้ชื่อเพลส ได้แก่ red1 green1 yellow1 และ safe1 แสดงถึงไฟแดง ไฟเขียว ไฟเหลืองและสถานะที่ปลอดภัยของเสาไฟจราจรต้นที่ 1 ในทำนองเดียวกันเพลส red2 green2 yellow2 และ safe2 แสดงถึงไฟแดง ไฟเขียว ไฟเหลืองและสถานะที่ปลอดภัยของเสาไฟจราจรต้นที่ 2 ในรูปที่ 2.4 (ก) แสดงตัวอย่างของแบบจำลองเพทรีเน็ตและ (ข) มาร์กกิงทั้งหมดของระบบไฟจราจรสองต้น โดยมีคำจำกัดความที่เป็นทางการมีการกำหนด  $PN = (P, T, F, V, M_0)$  ดังนี้

-  $P = \{\text{green1, green2, red1, red2, safe1, safe2, yellow1, yellow2}\}$

-  $T = \{t1, t2, t3, t4, t5, t6\}$

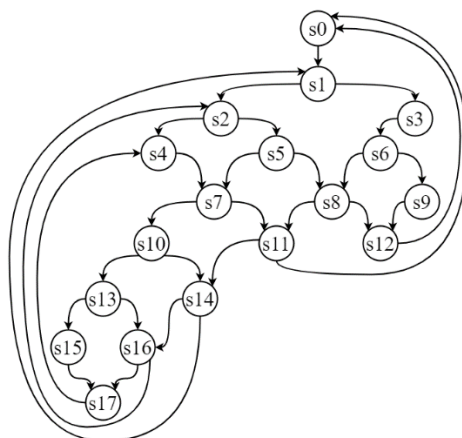
-  $F = \{(\text{safe1, t1}), (\text{red1, t1}), (\text{green1, t2}), (\text{yellow1, t3}), (\text{safe2, t4}), (\text{red2, t4}), (\text{green2, t5}), (\text{yellow2, t6}), (t1, \text{green1}), (t1, \text{safe2}), (t2, \text{yellow1}), (t3, \text{red1}), (t4, \text{green2}), (t4, \text{safe1}), (t5, \text{yellow2}), (t6, \text{red2})\}$

- $V = \{((\text{safe1}, t1), 1), ((\text{red1}, t1), 1), ((\text{green1}, t2), 1), ((\text{yellow1}, t3), 1), ((\text{safe2}, t4), 1), ((\text{red2}, t4), 1), ((\text{green2}, t5), 1), ((\text{yellow2}, t6), 1), ((t1, \text{green1}), 1), ((t1, \text{safe2}), 1), ((t2, \text{yellow1}), 1), ((t3, \text{red1}), 1), ((t4, \text{green2}), 1), ((t4, \text{safe1}), 1), ((t5, \text{yellow2}), 1), ((t6, \text{red2}), 1)\}$
- $M_0 = (0, 0, 1, 1, 1, 0, 0, 0)$  ซึ่งแสดงจำนวนโทเคนในเพลส (green1, green2, red1, red2, safe1, safe2, yellow1, yellow2)



รูปที่ 2.4 ตัวอย่างแบบจำลองเพทรีเน็ตสำหรับระบบไฟจราจรสองต้น

ในทางปฏิบัติ การเข้าถึงมาร์กกิง (Reachable marking) ของแบบจำลองเพทรีเน็ต สามารถสร้างเป็นกราฟการเข้าถึง (Reachability graph) ซึ่งถือว่าเป็นสถานะของระบบเช่นกัน ดังนั้น มาร์กกิง  $M_0, M_1, M_2, \dots, M_n$  สามารถแสดงเป็นสถานะของระบบได้เช่นกัน คือ  $s_0, s_1, s_2, \dots, s(n)$  โดยที่  $n$  คือจำนวนของสถานะในระบบที่สนใจ ตัวอย่างเช่น ระบบไฟจราจรสองต้น มาร์กกิง  $M_0$  แสดงถึงสถานะ  $s_0$  คือ พฤติกรรมการทำงานที่เพลส “red1” “red2” และ “safe1” เป็นจริง ในทางกลับกัน หากเพลส “red1” “red2” และ “safe1” เป็นจริง ทำให้ระบบอยู่ในสถานะเริ่มต้น คือ สถานะ  $s_0$  เป็นต้น ซึ่งสถานะที่เป็นไปได้ทั้งหมดของระบบ สามารถแสดงได้ด้วยปริภูมิสถานะซึ่งเป็นชุดของสถานะทั้งหมด ดังในรูปที่ 2.5 ทำให้สามารถวิเคราะห์และตรวจสอบพฤติกรรมของระบบ โดยที่คุณสมบัติเชิงพฤติกรรมเป้าหมายสามารถเขียนลงในสูตรตรรกศาสตร์เชิงเวลาแบบลำดับหรือแอลทีแอล โดยจะมีการกล่าวถึงในลำดับถัดไป



รูปที่ 2.5 ตัวอย่างกราฟปริภูมิสถานะของระบบไฟจราจรสองต้น

ดังนั้น ในงานวิจัยนี้ได้นำแบบจำลองเพทรีเน็ตแบบธรรมดา ซึ่งไม่มีเรื่องเวลาเข้ามาเกี่ยวข้อง ใช้เป็นแบบจำลองการทำงานของระบบที่เราสนใจ เพื่อนำมาสกัดเป็นข้อมูลสำหรับการเรียนรู้ โดยตัวอย่างของระบบเช่น ระบบเครื่องรับจ่ายเงินอัตโนมัติ หรือระบบสัญญาณไฟจราจร เป็นต้น

### 2.1.1.3 ปริภูมิสถานะ (State space)

ปริภูมิสถานะ (State space) หมายถึง ชุดของข้อมูลที่เป็นไปได้ทั้งหมดของระบบ ซึ่งสามารถนำมาใช้ประโยชน์สำหรับวิเคราะห์พฤติกรรมการทำงานทั้งหมดของระบบได้ ตัวแปรสถานะเป็นตัวแปรภายในระบบที่แสดงสถานะทั้งหมดในทุกช่วงเวลา โดยทั่วไปแล้ว การกำหนดแบบจำลองปริภูมิสถานะ คือ การแสดงพฤติกรรมทางกายภาพของระบบที่เขียนในรูปแบบทางคณิตศาสตร์ ประกอบด้วย ชุดของข้อมูลนำเข้า ข้อมูลส่งออกและสถานะที่เกี่ยวข้อง โดยการกำหนดเป็นสมการเชิงอนุพันธ์อันดับหนึ่ง แบบจำลองปริภูมิสถานะสามารถนำไปใช้ประโยชน์ในหลายสาขา ตัวอย่างเช่น

1) วิศวกรรมควบคุม (Control engineering) [23] แบบจำลองปริภูมิสถานะแสดงโดยการกำหนดสมการเชิงอนุพันธ์เพื่อวิเคราะห์พฤติกรรมของระบบควบคุม

2) เศรษฐมิติ (Econometrics) [24] แบบจำลองปริภูมิสถานะถูกสร้างโดยใช้วิธีการทางทฤษฎีสถิติทางคณิตศาสตร์สำหรับค้นหาตัวประมาณการที่เหมาะสม เพื่อนำไปวิเคราะห์ข้อมูลต่างๆ เช่น จุดเปลี่ยนของวัฏจักรธุรกิจ การประมาณการผลิตภัณฑ์รวมในประเทศหรือจีดีพี

3) วิทยาการคอมพิวเตอร์ (Computer science) [25] แบบจำลองปริภูมิสถานะถูกสร้างโดยใช้ฟังก์ชันเพื่ออธิบายสถานะทั้งหมดของระบบ ปริภูมิสถานะที่ถูกสร้างจะมีลักษณะเฉพาะที่แสดงปริภูมิสถานะของแต่ละระบบ ซึ่งสามารถแปลงให้อยู่ในรูปแบบของกราฟแบบมีทิศทาง (Directed graph) ได้ แต่ละสถานะจะแสดงเป็นจุดยอด (Vertex) และมีเส้นเชื่อม (Arch) แบบมีทิศทางเชื่อม



โดยทั่วไป การแจกแจงปริภูมิสถานะของโมเดลเซ็ทกิงถูกเก็บไว้ในหน่วยความจำ ที่อยู่ในรูปแบบของกราฟ ถ้าหากระบบที่มีขนาดใหญ่และมีความซับซ้อนก็จะทำให้มีสถานะจำนวนมากมายมหาศาล ทำให้สิ้นเปลืองพื้นที่หน่วยความจำและอาจส่งผลให้เกินขีดความสามารถของโมเดลเซ็ทกิง ดังนั้น วิธีการค้นหาไปที่สถานะโดยการมองไปข้างหน้า (Lookahead) โดยเริ่มจากสถานะเริ่มต้นแล้วค่อยๆ ค้นหาสถานะถัดไปจนพบสถานะเป้าหมายสามารถช่วยแก้ปัญหาการค้นหาของหน่วยความจำได้ โดยหลักการทำงาน คือ การแจกแจงปริภูมิสถานะไปข้างหน้าเพื่อค้นหาสถานะเป้าหมาย โดยสถานะที่ไม่เกี่ยวข้องจะไม่ถูกเก็บหรือตัดทิ้งไป ทำให้ใช้หน่วยความจำน้อยลง เช่น การแจกแจงปริภูมิสถานะแบบการลดบางส่วน (Partial reduction) หรือการตัดแต่งกิ่งปริภูมิสถานะ (State space pruning) เป็นต้น ดังนั้น ในงานวิจัยนี้ใช้โครงสร้างของกราฟปริภูมิสถานะสำหรับแสดงการแจกแจงของปริภูมิสถานะของระบบที่มีการทำงานแบบจำกัด เพื่อป้องกันระบบที่มีการทำงานแบบวนซ้ำ

#### 2.1.1.4 ตรรกศาสตร์เชิงเวลาแบบลำดับ (Linear temporal logic)

ตรรกศาสตร์เชิงเวลาแบบลำดับหรือแอลทีแอล (Linear temporal logic หรือ LTL) ได้รับการนำเสนอครั้งแรกในปี 1977 โดย อเมียร์ นูว์ลี (Amir Pnueli) [26] เป็นตรรกศาสตร์เชิงเวลาแบบลำดับที่แสดงพฤติกรรมของระบบที่มีเงื่อนไขของเวลาเข้ามาเกี่ยวข้อง โดยสามารถเขียนออกมาในรูปแบบของตรรกศาสตร์ประพจน์การให้เหตุผลแบบนิรนัยที่มีค่าความจริงเป็น “จริง” หรือ “เท็จ” เท่านั้น ดังนั้น สูตรแอลทีแอลถูกนำมาใช้เพื่ออธิบายพฤติกรรมของระบบซอฟต์แวร์สำหรับการทวนสอบพฤติกรรมการทำงานของระบบ ซึ่งการตีความโดยทั่วไปจะเกี่ยวข้องกับเวลาเริ่มต้นเสมอและดำเนินต่อไปจนถึงอนาคตประกอบด้วยประพจน์เดี่ยวหรือเอพี (Atomic proposition หรือ AP) ที่มีตัวเชื่อมแต่ละประพจน์และตัวดำเนินการเชิงเวลารวมอยู่ด้วยกันได้ โดยตัวเชื่อมประพจน์ มีดังต่อไปนี้

- 1) นิเสธ (Negation) เขียนแทนด้วยสัญลักษณ์  $\neg$
- 2) และ (Conjunction) เขียนแทนด้วยสัญลักษณ์  $\wedge$
- 3) หรือ (Disjunction) เขียนแทนด้วยสัญลักษณ์  $\vee$
- 4) การให้เหตุผล (Implication) เขียนแทนด้วยสัญลักษณ์  $\rightarrow$

ตัวดำเนินการเชิงเวลาที่มักใช้ในตรรกศาสตร์นี้ ได้แก่

- 1) เสมอ (Always) เขียนแทนด้วยสัญลักษณ์ G
- 2) ถัดไป (Next) เขียนแทนด้วยสัญลักษณ์ X
- 3) ในที่สุด (Finally) เขียนแทนด้วยสัญลักษณ์ F
- 4) จนกระทั่ง (Until) เขียนแทนด้วยสัญลักษณ์ U

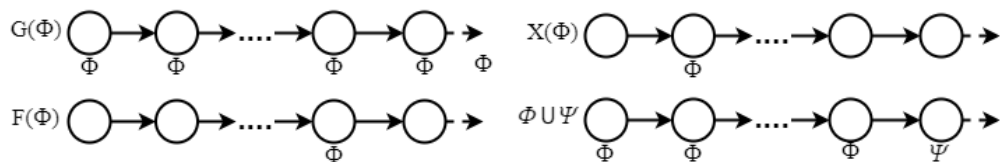
การตีความของตรรกศาสตร์ที่มีตัวดำเนินการเชิงเวลาในงานวิจัยนี้สนใจเวลาจากจุดเริ่มต้น และต่อเนื่องไปยังอนาคต โดยไม่สนใจกรณีในอดีต ดังนั้น แบบจำลองเวลาแบบเส้นตรงของประพจน์ที่เขียนด้วยตัวดำเนินการเชิงเวลา สามารถแสดงในรูปที่ 2.7 ความหมายของแต่ละตัวดำเนินการเชิงเวลาที่ถูกใช้ในงานวิจัยนี้ โดยกำหนดให้  $\Phi$  และ  $\Psi$  คือ สูตรแอลทีแอล ซึ่งสามารถตีความได้ดังต่อไปนี้

$G(\Phi)$  หมายถึง การทวนสอบว่า  $\Phi$  จะเป็นจริงเสมอในทุกช่วงเวลา

$X(\Phi)$  หมายถึง การทวนสอบว่า  $\Phi$  จะเป็นจริงในเวลาถัดไปเท่านั้น

$F(\Phi)$  หมายถึง การทวนสอบว่า  $\Phi$  จะเป็นจริงในที่สุด

$\Phi \cup \Psi$  หมายถึง การทวนสอบว่า  $\Phi$  จะเป็นจริงเสมอ จนกระทั่ง  $\Psi$  เป็นจริงในที่สุด



รูปที่ 2.7 แบบจำลองเวลาแบบเส้นตรงของประพจน์ที่เขียนด้วยตัวดำเนินการเชิงเวลา [26]

วิธีการเขียนสูตรแอลทีแอลที่ถูกต้องจะต้องใช้ตัวดำเนินการเชิงเวลาและตัวเชื่อมประพจน์ที่กล่าวมาข้างต้น เพื่อให้ได้สูตรแอลทีแอลที่เหมาะสมหรือดับเบิลยูเอฟแอล (Well-formed LTL หรือ WFL) เมื่อกำหนดประพจน์  $prop \in AP$  แล้ว  $prop$  ถือว่าเป็น WFL ดังนั้น ถ้า  $\Phi$  และ  $\Psi$  คือ WFL แล้ว  $(\Phi)$ ,  $\neg\Phi$ ,  $(\Phi \wedge \Psi)$ ,  $(\Phi \vee \Psi)$ ,  $(\Phi \rightarrow \Psi)$ ,  $G(\Phi)$ ,  $F(\Phi)$ ,  $X(\Phi)$  และ  $(\Phi \cup \Psi)$  ก็คือ WFL เช่นกัน

การตีความของสูตรแอลทีแอล ถูกตีความในรูปแบบของ  $\langle \pi, i \rangle$  ซึ่ง  $\pi$  คือ โมเดลหรือสถานะ ณ เวลา  $i$  โดยปกติแล้ว การทวนสอบระบบจะเริ่มจากสถานะเริ่มต้น คือ  $i = 1$  ดังนั้นจะสามารถเขียนสมการได้เป็น  $\langle \pi, 1 \rangle$  เมื่อต้องการทวนสอบระบบด้วยการกำหนดสูตรแอลทีแอล สามารถเขียนให้อยู่ในสมการ  $\langle \pi, i \rangle \models (\Phi)$  หมายความว่า “ $\langle \pi, i \rangle$  Satisfies  $\Phi$ ” ก็ต่อเมื่อ  $\pi$  ณ ช่วงเวลา  $i$  เมื่อพิจารณาตามเงื่อนไขของตัวดำเนินการเวลาทำให้  $\Phi$  มีค่าเป็นจริง ซึ่งในทางตรงกันข้าม เมื่อพิจารณาที่มีค่าเป็นเท็จ ก็จะหมายความว่า  $\langle \pi, i \rangle \not\models (\Phi)$  หมายความว่า “ $\langle \pi, i \rangle$  ไม่ Satisfies  $\Phi$ ” ซึ่งเครื่องหมาย  $\models$  (Phi) คือ สัญลักษณ์ความสัมพันธ์ด้านความพึงพอใจ การตีความสูตรแอลทีแอลว่ามีค่าความจริงเป็นจริงหรือเท็จ สามารถกำหนดได้จากนิยาม [27] ดังต่อไปนี้

$$\langle \pi, i \rangle \models prop \quad \text{iff} \quad prop \in AP$$

$$\begin{aligned}
\langle \pi, i \rangle \models (\Phi) & \quad \text{iff} \quad \langle \pi, i \rangle \models (\Phi) \\
\langle \pi, i \rangle \models \neg(\Phi) & \quad \text{iff} \quad \langle \pi, i \rangle \not\models (\Phi) \\
\langle \pi, i \rangle \models (\Phi \wedge \Psi) & \quad \text{iff} \quad \langle \pi, i \rangle \models (\Phi) \text{ และ } \langle \pi, i \rangle \models (\Psi) \\
\langle \pi, i \rangle \models (\Phi \vee \Psi) & \quad \text{iff} \quad \langle \pi, i \rangle \models (\Phi) \text{ หรือ } \langle \pi, i \rangle \models (\Psi) \\
\langle \pi, i \rangle \models (\Phi \rightarrow \Psi) & \quad \text{iff} \quad \text{if } \langle \pi, i \rangle \models (\Phi) \text{ then } \langle \pi, i \rangle \models (\Psi) \\
\langle \pi, i \rangle \models G(\Phi) & \quad \text{iff} \quad \forall j \geq i: \langle \pi, j \rangle \models (\Phi) \\
\langle \pi, i \rangle \models F(\Phi) & \quad \text{iff} \quad \exists j \geq i: \langle \pi, j \rangle \models (\Phi) \\
\langle \pi, i \rangle \models X(\Phi) & \quad \text{iff} \quad \langle \pi, i+1 \rangle \models (\Phi) \\
\langle \pi, i \rangle \models (\Phi \cup \Psi) & \quad \text{iff} \quad \exists j \geq i: \langle \pi, j \rangle \models (\Psi) \text{ และ } \forall i \leq k < j: \langle \pi, k \rangle \models (\Phi)
\end{aligned}$$

จากที่กล่าวมาข้างต้น ทำให้สามารถกำหนดคุณสมบัติที่ใช้ในการทวนสอบการทำงานของระบบว่ามีคุณสมบัติการทำงานที่จำลองโดยแบบจำลองเพทรีเน็ต โดยใช้สูตรแอลที่แอลที่เหมาะสม WFL โดยคุณสมบัติพื้นฐานที่ที่ใช้ในการทวนสอบพฤติกรรมการทำงานของระบบ

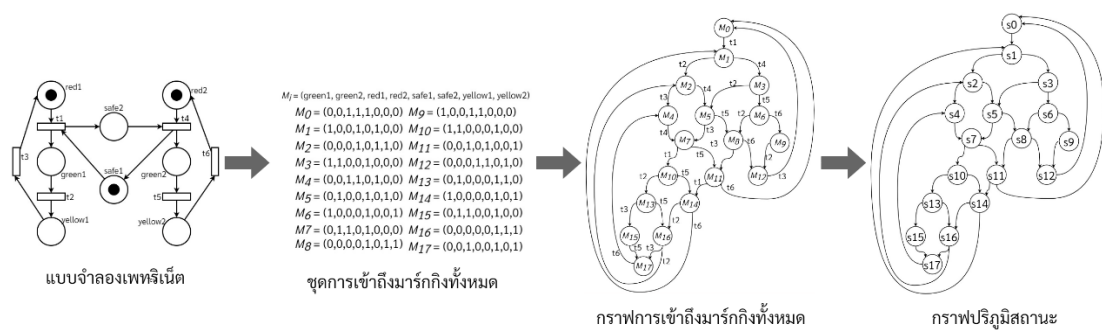
โดยทั่วไป การตรวจสอบความถูกต้องของการทำงานของระบบซอฟต์แวร์สามารถตรวจสอบได้จากคุณสมบัติด้านความปลอดภัยและความดำเนินชีวิต [26] ซึ่งทั้ง 2 คุณสมบัตินี้เป็นพื้นฐานของคุณสมบัติอื่นๆ [28] เช่น คุณสมบัติด้านความสัมพันธ์ เป็นต้น ซึ่งสามารถเขียนออกมาโดยใช้สูตรแอลที่แอลได้ [29] โดยมีรายละเอียด ดังนี้

- 1) คุณสมบัติด้านความปลอดภัย (Safety property) เป็นการทวนสอบให้แน่ใจว่าเหตุการณ์หรือสถานการณ์ที่ไม่ดี ไม่เกิดขึ้นเสมอตลอดการทำงานของระบบ กล่าวคือ พฤติกรรมการทำงานของระบบจะทำงานโดยไม่มีข้อผิดพลาดเกิดเสมอ ดังนั้น สามารถเขียนแทนโดยใช้สูตรแอลที่แอลโดยการใส่เครื่องหมายนิเสธหน้าเหตุการณ์ที่ไม่ต้องการให้เกิด สามารถเขียนสูตรแอลที่แอล คือ  $G\neg(p)$  โดยที่  $p$  คือเหตุการณ์ที่ไม่ดีจะไม่เกิดขึ้นเสมอตลอดการทำงานของระบบ
- 2) คุณสมบัติด้านความดำเนินชีวิต (Liveness property) เป็นการทวนสอบให้แน่ใจว่าระบบจะมีสถานการณ์ที่ต้องการเกิดขึ้นในที่สุด ดังนั้น สามารถเขียนแทนโดยใช้สูตรแอลที่แอล คือ  $F(q)$  โดยที่  $q$  คือ พฤติกรรมที่ต้องการจะเกิดขึ้นในที่สุด
- 3) คุณสมบัติด้านความสัมพันธ์ (Correlation property) เป็นการทวนสอบให้แน่ใจว่าในที่สุดแล้วเมื่อมีระบบทำงานใดงานหนึ่งแล้ว งานที่มีความสัมพันธ์กับงานนั้นจะทำงานในที่สุด ดังนั้น สามารถเขียนแทนโดยใช้สูตรแอลที่แอล คือ  $F(p) \rightarrow F((p) \wedge F(q))$  โดยที่



$p$  คือ งานที่ระบบทำเป็นงานแรกและ  $q$  คือ งานที่มีความสัมพันธ์กับ  $p$  ที่จะทำในที่สุด กล่าวคือ ในที่สุดแล้ว  $p$  จะทำงาน จากนั้น  $q$  จะทำงานในที่สุด

เพื่อให้เข้าใจวิธีการนำเพทรีเน็ตมาจำลองการทำงานของระบบและวิธีการทวนสอบการทำงานที่จำลองโดยเพทรีเน็ต ในวิทยานิพนธ์นี้ได้แสดงตัวอย่างของระบบสัญญาณไฟจราจรที่มีเสาไฟจราจรสองต้น ประกอบด้วย สัญญาณ ไฟแดง ไฟเขียวและไฟเหลือง ตามลำดับ และเพื่อแสดงความสอดคล้องกันระหว่างแบบจำลองเพทรีเน็ตแสดงในรูปที่ 2.4 และปริภูมิสถานะแสดงในรูปที่ 2.5 สามารถแสดงเป็นลำดับได้ดังรูปที่ 2.8



รูปที่ 2.8 ลำดับการนำแบบจำลองเพทรีเน็ตมาสร้างเป็นกราฟปริภูมิสถานะ

จากรูปที่ 2.8 คือ ลำดับของการนำแบบจำลองเพทรีเน็ตมาสร้างเป็นกราฟปริภูมิสถานะ โดยมีรายละเอียด ดังต่อไปนี้

- 1) เริ่มจากนำแบบจำลองเพทรีเน็ตที่จำลองการทำงานของระบบที่สนใจมาทำการเข้าถึงมาร์กกิงทั้งหมด
- 2) เมื่อได้ชุดของการเข้าถึงมาร์กกิงทั้งหมด สามารถนำมาวาดเป็นกราฟของการเข้าถึงมาร์กกิงทั้งหมด โดยแต่ละโหนดจะกำกับเป็นชื่อมาร์กกิง
- 3) กำหนดให้แต่ละมาร์กกิงเท่ากับสถานะ เช่น  $M_0$  เท่ากับ  $S_0$   $M_1$  เท่ากับ  $s_1$  และ  $M_{(i)}$  เท่ากับ  $s(i)$  โดยที่  $i$  คือจำนวนมาร์กกิงทั้งหมด จากนั้น นำแต่ละสถานะมาวาดเป็นกราฟปริภูมิสถานะได้ โดยแต่ละสถานะเชื่อมต่อกันเช่นเดียวกันกับมาร์กกิง ซึ่งไม่สนใจทรานซิชันที่อยู่บนเส้นเชื่อม ก็จะทำให้ได้เป็นกราฟปริภูมิสถานะของแบบจำลองเพทรีเน็ตสามารถแสดงเป็นขั้นตอนได้ดังนี้
  - 3.1) ระบบเริ่มทำงาน โดยดูจากมาร์กกิงเริ่มต้น  $M_0$  มีเพลสที่มีโทเคนบรรจุได้แก่ red1 safe1 และ red2 แสดงให้เห็นว่าระบบอยู่ในสถานะเริ่มต้น  $s_0$
  - 3.2) จากนั้น ทรานซิชัน  $t_1$  ถูกเปิดใช้งานและยิงโทเคนจาก red1 ไปยัง green1 และ safe2 ซึ่งทำให้มาร์กกิงถูกเปลี่ยนจาก  $M_0$  ไปยัง  $M_1$  หรือสถานะ  $s_0$  ไปยัง  $s_1$

3.3) จากนั้น ทำให้ทรานซิชัน  $t_2$  และ  $t_4$  ถูกเปิดใช้งานพร้อมกัน เมื่อทรานซิชัน  $t_2$  เปลี่ยนเป็นสถานะยิง ทำให้โทเคนจากเพลส green1 ถูกยิงออกไปยังเพลส yellow1 และเมื่อลองสังเกตดูพบว่ามีเพลสที่มีโทเคนบรรจุอยู่ได้แก่ red2 safe2 และ yellow1 ซึ่งก็คือมาร์กกิง  $M_2$  หรือสถานะ  $s_2$  และเมื่อทรานซิชัน  $t_4$  เปลี่ยนเป็นสถานะยิง ทำให้โทเคนจากเพลส red2 และ safe2 ถูกยิงออกไปยังเพลส green2 และ safe1 ซึ่งก็คือมาร์กกิง  $M_3$  หรือสถานะ  $s_3$  ดังนั้น จากข้อ 2) มาร์กกิง  $M_1$  หรือสถานะ  $s_1$  สามารถเปลี่ยนไปยัง  $M_2$  หรือสถานะ  $s_2$  หรือ  $M_3$  หรือสถานะ  $s_3$  ได้เช่นกัน

3.4) เมื่อทำแบบนี้ไปจนครบทุกมาร์กกิง ก็ทำให้ได้เป็นกราฟปริภูมิสถานะของระบบ

เมื่อได้กราฟปริภูมิสถานะเรียบร้อยแล้ว สามารถอธิบายการทำงานของระบบสัญญาณไฟจราจรสองต้นที่แสดงให้เห็นถึงพฤติกรรมที่เสาไฟจราจรเป็นสีเขียวพร้อมกัน ได้ดังต่อไปนี้

- 1) เสาไฟจราจรต้นที่หนึ่งเริ่มทำงานก่อน โดยเริ่มจาก red1 green1 และ yellow1 ตามลำดับ
- 2) จากนั้น เข้าสู่สถานะ safe2 ซึ่งจะสลับไปให้เสาไฟจราจรต้นที่สองทำงาน โดยเริ่มจาก red2 green2 และ yellow2 ตามลำดับ
- 3) เมื่อลองสังเกตดูพฤติกรรมในช่วงเวลาที่ทรานซิชัน  $t_1$  ถูกเปิดใช้งาน ทำให้โทเคนจาก red1 ถูกยิงไปยัง green1 และ safe2 จากนั้นจะทำให้ทรานซิชัน  $t_4$  ถูกเปิดใช้งานและยิงโทเคนจาก red2 ไปยัง green2 ทำให้เกิดเหตุการณ์ที่ทำให้ไฟจราจรทั้ง 2 ต้นเป็นสีเขียวพร้อมกัน แสดงได้ตั้งมาร์กกิง  $M_3$  และ  $M_{10}$  หรือสถานะ  $s_3$  และ  $s_{10}$  ซึ่งหมายความว่าการทำงานของระบบสัญญาณไฟจราจรนี้มีข้อผิดพลาดเกิดขึ้น

จากตัวอย่างการจำลองการทำงานของระบบสัญญาณไฟจราจรสองต้นโดยการใช้เพทรีเน็ต จะสังเกตเห็นว่า มาร์กกิงเริ่มต้นมีมากกว่า 1 เพลส คือ เพลส safe1 red1 และ red2 โดยแต่ละมาร์กกิงก็สามารถมีได้มากกว่า 1 เพลสเช่นเดียวกัน แสดงให้เห็นว่า ในขณะที่นั้นมีสถานะใดเกิดขึ้นภายในระบบ ตัวอย่างเช่น เมื่อสังเกต  $M_3$  หรือสถานะ  $s_3$  จะเห็นได้ว่า สัญญาณ green1 และสัญญาณ green2 และ safe1 เกิดขึ้น เป็นต้น จากนั้น เมื่อได้หามาร์กกิงทั้งหมด ก็จะได้เป็นปริภูมิสถานะทั้งหมดของระบบ ดังแสดงในรูปที่ 2.5 ซึ่งต่อจากนี้จะขอกล่าวถึงเพียงสถานะเท่านั้น จากนั้น จะสามารถทวนสอบคุณสมบัติการทำงานของระบบสัญญาณไฟจราจรนี้เป็นไปตามที่คาดหวังหรือไม่ หากเป็นไปตามคาดหวังจะได้ผล “พอใจ” (Satisfied) แต่ในทางตรงข้ามจะได้ผล “ไม่พอใจ” (Not satisfied) ซึ่งสามารถนำแบบจำลองเพทรีเน็ตที่จำลองการทำงานของระบบสัญญาณไฟจราจรมาทวน

สอบระบบได้โดยการเขียนสูตรแอลทีแอล แสดงเป็นตัวอย่างของการทวนสอบคุณสมบัติพื้นฐาน โดยมีรายละเอียดการทวนสอบดังนี้

### 1) ด้านความปลอดภัย

1.1) ต้องการทวนสอบว่า สัญญาณไฟเขียวจะไม่เกิดขึ้นพร้อมกันเสมอ สามารถเขียนเป็นสูตรแอลทีแอล คือ  $G(\text{green1} \wedge \text{green2})$  หมายความว่า สัญญาณไฟเขียวของเสาไฟจราจรของทั้งสองต้นเกิดพร้อมกัน เป็นสิ่งที่คาดหวังจะไม่ให้เกิด โดยการทวนสอบจะทำการค้นหาว่ามีมาร์กกิงใดบ้างที่มีเพลส green1 และ green2 บรรจุโทเคน 1 อัน จากการสังเกตพบว่า สถานะ s3 และ s10 มีเพลส green1 และ green2 เกิดขึ้นพร้อมกัน บนเส้นทางการดำเนินงานในปริภูมิสถานะอย่างน้อยหนึ่งเส้นทาง ดังนั้น ผลการทวนสอบนี้เป็นที่พอใจ

### 2) ด้านความดำเนินชีวิต

2.1) ต้องการทวนสอบว่า สัญญาณไฟจราจรต้นที่ 1 จะเป็นสีแดงในที่สุด สามารถเขียนเป็นสูตรแอลทีแอลได้ คือ  $F(\text{red1})$  โดยการทวนสอบจะทำการค้นหาทุกเส้นทางการดำเนินงานในปริภูมิสถานะ ว่ามีมาร์กกิงหรือสถานะใดที่มี red1 บรรจุโทเคน 1 อันหรือไม่ ซึ่งในตัวอย่างนี้จะพบว่า สถานะ s0 s4 s7 s11 s15 และ s17 จะพบบนทุกเส้นทางการดำเนินงานในปริภูมิสถานะ ดังนั้น ผลการทวนสอบนี้เป็นที่พอใจ

2.2) ต้องการทวนสอบว่าสัญญาณไฟจราจรต้นที่ 1 เป็นสีแดงและสัญญาณไฟจราจรต้นที่ 2 เป็นสีเหลืองพร้อมกันในที่สุด สามารถเขียนเป็นสูตรแอลทีแอล คือ  $F(\text{red1} \wedge \text{yellow2})$  โดยการทวนสอบจะทำการค้นหาว่ามีมาร์กกิงใดบ้างที่มีเพลส red1 และ yellow2 บรรจุโทเคน 1 อัน จากการสังเกตจะพบ สถานะ s17 จะไม่พบบนบางเส้นทางการดำเนินงานในปริภูมิสถานะ เนื่องจาก อาจมีสถานการณ์ที่เสาไฟจราจรต้นที่ 2 เปลี่ยนสีจากสีแดงเป็นสีเขียว สีเหลืองและกลับมาเป็นสีแดงอีกครั้ง โดยที่เสาไฟจราจรต้นที่ 1 ยังคงเป็นสีเขียวอยู่ ซึ่งทำให้สถานการณ์ที่ถูกระบุโดยสูตรแอลทีแอลอาจไม่เกิดขึ้นในบางเส้นทางการดำเนินงานนั่นเอง ดังนั้น ผลการทวนสอบนี้เป็นที่พอใจ

### 3) ด้านความสัมพันธ์

3.1) ต้องการทวนสอบว่า เมื่อสัญญาณไฟจราจรต้นที่ 2 เป็นสีแดง แล้วในที่สุดก็จะถูกเปลี่ยนเป็นสีเหลือง สามารถเขียนเป็นสูตรแอลทีแอล คือ  $F(\text{red2}) \rightarrow F((\text{red2}) \wedge F(\text{yellow2}))$  โดยการทวนสอบจะทำการค้นหาทุกเส้นทางการดำเนินงานในปริภูมิสถานะ ว่ามีสถานะใดที่มี red2 บรรจุโทเคน 1 อันหรือไม่ ซึ่งในตัวอย่างนี้จะพบว่ามี

สถานะ s0 s1 s2 s4 s9 และ s12 แล้วจะมีสถานะ s6 s8 s11 s14 s16 และ s17 ที่มี yellow2 บรรจุโทเคน 1 อัน เมื่อพิจารณาเส้นทางดำเนินงานในปริภูมิสถานะแล้วจะพบว่า เมื่อสถานะ s0 s1 s2 s4 s9 หรือ s12 เป็นจริงแล้ว สถานะ s6 s8 s11 s14 s16 และ s17 จะพบบนทุกเส้นทางดำเนินงาน ดังนั้น ผลการทวนสอบนี้จะเป็นที่พอใจ

3.2) ต้องการทวนสอบว่า เมื่อสัญญาณไฟจราจรต้นที่ 1 เป็นสีแดง แล้วจะถูกเปลี่ยนเป็นสีเหลืองในเวลาถัดมา สามารถเขียนเป็นสูตรแอลทีแอล คือ  $F(\text{red1}) \rightarrow F((\text{red1}) \wedge X(\text{yellow1}))$  โดยการทวนสอบจะทำการค้นหาทุกเส้นทางดำเนินงานในปริภูมิสถานะ ว่ามีสถานะใดที่มี red1 บรรจุโทเคน 1 อันหรือไม่ ซึ่งในตัวอย่างนี้จะพบว่า มีสถานะ s0 s4 s7 s11 s15 และ s17 แล้วจะมีสถานะ s2 s5 s8 s12 s13 และ s16 ที่มี yellow1 บรรจุโทเคน 1 อัน เมื่อพิจารณาเส้นทางดำเนินงานในปริภูมิสถานะแล้วจะพบว่า มีบางเส้นทาง เมื่อสถานะ s0 s4 s7 s11 s15 หรือ s17 เป็นจริงแล้ว สถานะ s2 s5 s8 s12 s13 และ s16 จะไม่พบในบางเส้นทางดำเนินงาน ดังนั้น ผลการทวนสอบนี้จะเป็นที่ไม่พอใจ

นอกจากนี้ ตัวดำเนินการเชิงเวลาประเภท Until หรือ “จนกระทั่ง” ถูกใช้ทวนสอบให้แน่ใจว่า เมื่อมีการส่งคำร้องขอแล้ว การร้องขอนั้นจะถูกส่งไปเรื่อย ๆ จนกระทั่งมีงานใดๆ มาตอบรับการร้องขอนั้น จากนั้นคำร้องขอนั้นถึงจะหยุดได้ สามารถเขียนแทนโดยใช้สูตรแอลทีแอล คือ request U response โดยที่ request คือ งานที่ส่งสัญญาณคำร้องขอ ที่จะส่งสัญญาณไปเรื่อย ๆ จนกระทั่งได้รับคำตอบกลับจาก response จากนั้น request ถึงจะหยุดส่งสัญญาณได้

อย่างไรก็ตาม การทำงานของระบบที่มีขนาดใหญ่อาจพบโทเคนได้ไม่จำกัด ทำให้เฟลสสามารถปรากฏอยู่ในสถานะได้หลายสถานะ ดังนั้น จะส่งผลให้การทวนสอบจะมีความยุ่งยากและซับซ้อนมากยิ่งขึ้น

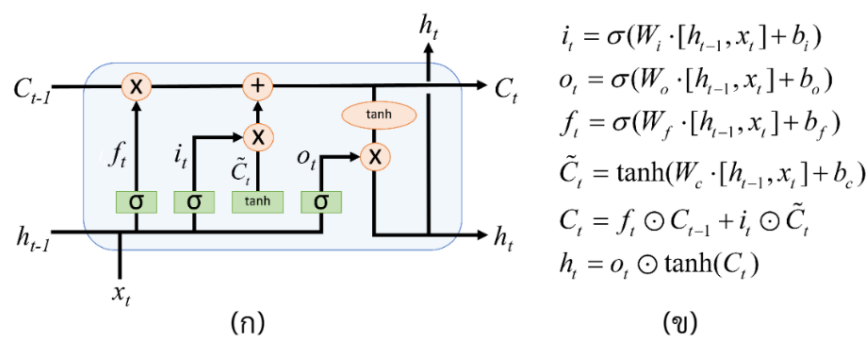
## 2.1.2 ทฤษฎีที่เกี่ยวข้องกับแบบจำลองแอลเอสทีเอ็มและการจำแนกแบบหลายป้าย

ในส่วนนี้นำเสนอทฤษฎีพื้นฐานของแบบจำลองแอลเอสทีเอ็ม ซึ่งประกอบด้วย โครงข่ายความจำระยะสั้นแบบยาวหรือแอลเอสทีเอ็ม การจำแนกแบบหลายป้ายและกลยุทธ์การหาขีดแบ่ง

### 2.1.2.1 โครงข่ายความจำระยะสั้นแบบยาวหรือแอลเอสทีเอ็ม (Long short-term memory network หรือ LSTM)

โครงข่ายความจำระยะสั้นแบบยาวหรือแอลเอสทีเอ็ม (Long short-term memory network หรือ LSTM) [5] ถูกนำเสนอโดยไฮโรเทอร์และซมิคูเบอร์ (Hochreiter and

Schmidhuber) ในปี 1997 ซึ่งเป็นโครงข่ายประสาทเทียม (Artificial neural network) ประเภทหนึ่ง que พัฒนามาจากโครงข่ายแบบวนซ้ำ (Recurrent neural network) เพื่อแก้ปัญหาการเรียนรู้ลำดับข้อมูลที่มีความยาวมากๆ เช่น การรู้จำคำพูด (Speech recognition) [30] การจำแนกข้อความ (Text classification) [10] และการเรียนรู้ไวยากรณ์ (Grammar learning) [31] เป็นต้น ซึ่งเมื่อผ่านการคำนวณแต่ละลำดับจะทำให้มีเกรเดียนต์เล็กลงเรื่อย ๆ จนมีค่าเท่ากับ 0 ซึ่งจะทำให้ค่าน้ำหนัก (Weight) ไม่ถูกอัปเดต ส่งผลให้โครงข่ายแบบวนซ้ำไม่สามารถเรียนรู้ได้อย่างมีประสิทธิภาพ ซึ่งเรียกว่า ปัญหาการสูญหายของเกรเดียนต์ (Vanishing gradient problem) [32] โดยที่ แอลเอสทีเอ็ม ที่ถูกพัฒนาออกจะมีความสามารถในการจดจำและลืม โดยมีประตู (Gate) ที่ทำหน้าที่ควบคุมการไหลของข้อมูล โครงสร้างสถาปัตยกรรมและสมการแอลเอสทีเอ็ม [33] ในรูปที่ 2.9 (ก) โครงสร้างสถาปัตยกรรมของแอลเอสทีเอ็มและ (ข) สมการของแอลเอสทีเอ็ม



รูปที่ 2.9 โครงสร้างสถาปัตยกรรมของแอลเอสทีเอ็มและสมการของแอลเอสทีเอ็ม [33]

- 1) ประตูทางเข้า (Input gate หรือ  $i_t$ ) ทำหน้าที่พิจารณาว่า ข้อมูลที่รับเข้ามาใหม่ถูกนำไปอัปเดตเซลล์ปัจจุบัน ( $C_t$ ) หรือไม่ โดยใช้ฟังก์ชันซิกมอยด์ (Sigmoid function หรือ  $\sigma$ ) คำนวณจากข้อมูลนำเข้าใหม่และสถานะที่ซ่อนอยู่ก่อนหน้า ( $h_{t-1}$ ) และพิจารณาด้วยว่าจะอัปเดตด้วยค่าใด โดยนำข้อมูลนำเข้ามาคำนวณโดยผ่านฟังก์ชันไฮเพอร์โบลิคแทนเจนต์หรือแทนเจนต์ (Hyperbolic tangent หรือ  $\tanh$ ) เพื่อให้ได้เป็นเซลล์สถานะใหม่ ( $\tilde{C}_t$ )
- 2) ประตูลืม (Forget gate หรือ  $f_t$ ) ทำหน้าที่พิจารณาว่า จะลืมข้อมูลของเซลล์สถานะปัจจุบันเพื่อเตรียมพื้นที่ให้ข้อมูลนำเข้าที่รับเข้ามาใหม่หรือไม่ โดยใช้ฟังก์ชันซิกมอยด์ คำนวณจากข้อมูลนำเข้าใหม่และสถานะที่ซ่อนอยู่ก่อนหน้า ( $h_{t-1}$ ) ซึ่งจะเป็นผลลัพธ์ของการพิจารณาครั้งก่อนหน้า หากได้ค่า 0 หมายความว่า ให้ลบเซลล์สถานะปัจจุบันทิ้งไปและให้เก็บผลลัพธ์ที่อยู่ในสถานะที่ซ่อนอยู่ก่อนหน้าแทน แต่หากได้

- ค่า 1 หมายความว่า ให้จำเซลล์สถานะปัจจุบันและลืมข้อมูลจากสถานะที่ซ่อนอยู่ก่อนหน้า
- 3) การอัปเดตสถานะเซลล์ปัจจุบัน พิจารณาผลรวมของข้อมูลที่ได้จากประตูลืมและประตูทางเข้า โดยที่ ประตูลืมมีค่าเป็น 0 หมายความว่า ค่าสถานะเซลล์ก่อนหน้า ( $C_{t-1}$ ) จะไม่ถูกนำมาพิจารณาการอัปเดตเลย แต่ในทางตรงกันข้าม เซลล์ก่อนหน้าจะถูกนำมาพิจารณาการอัปเดต ส่วนของประตูทางเข้าจะเป็นการพิจารณาว่าเซลล์สถานะใหม่จะถูกนำมาใช้หรือไม่ ซึ่งหากว่า ค่าประตูทางเข้ามีค่าเป็น 1 หมายความว่า ให้นำเซลล์สถานะใหม่ที่ได้รับการพิจารณาจากข้อมูลที่รับเข้ามา จะถูกนำมาใช้ประกอบการอัปเดตสถานะเซลล์ปัจจุบัน แต่หากมีค่าเป็น 0 หมายความว่า สถานะเซลล์ใหม่จะไม่ถูกนำมาใช้อัปเดตสถานะเซลล์ปัจจุบันเลย
- 4) ประตูทางออก (Output gate หรือ  $o_t$ ) ทำหน้าที่พิจารณาว่า ผลลัพธ์ของสถานะที่ซ่อนอยู่ในปัจจุบัน ( $h_t$ ) มีค่าเท่าใด เพื่อเลือกว่าจะส่งค่าใดๆ ออกไปให้คำนวณในเวลาถัดไป  $t+1$  หรือไม่ โดยประตูทางออกใช้ฟังก์ชันซิกมอยด์คำนวณจากข้อมูลนำเข้าใหม่และสถานะที่ซ่อนอยู่ก่อนหน้า จากนั้น คำนวณหาผลลัพธ์ของสถานะที่ซ่อนอยู่ในปัจจุบัน โดยพิจารณาจากค่าของประตูทางออก หากมีค่าเป็น 0 จะทำให้สถานะที่ซ่อนอยู่ในปัจจุบันมีค่าเป็น 0 เช่นกัน หมายความว่า ไม่มีค่าใดๆ ถูกส่งออกไป ในขณะที่ หากประตูทางออกมีค่าเป็น 1 หมายความว่า มีการคำนวณจากการนำผลของประตูทางออกและผลลัพธ์ของสถานะเซลล์ปัจจุบันและส่งออกไปยังเวลาถัดไป

จากสมการใน รูปที่ 2.9 (ข) ประกอบด้วยตัวแปร ดังต่อไปนี้

$i_t$ คือ ประตูรับข้อมูล	$o_t$ คือ ประตูส่งออก
$f_t$ คือ ประตูลืม	$C_t$ คือ สถานะเซลล์ปัจจุบัน
$\tilde{C}_t$ คือ สถานะเซลล์ใหม่	$x_t$ คือ ข้อมูลนำเข้าที่เข้ามาในเซลล์ ณ เวลาปัจจุบัน
$h_{t-1}$ คือ สถานะที่ซ่อนอยู่ก่อนหน้า	$W$ คือ ค่าน้ำหนัก
$b$ คือ ค่าอคติ (Bias)	$\sigma$ คือ ฟังก์ชันซิกมอยด์
$\tanh$ คือ ฟังก์ชันไฮเพอร์โบลิกแทนเจนต์	$\odot$ คือ ตัวคูณ (Element-wise multiplication)

ผลการทำนายของโครงข่ายประสาทเทียมมักจะอยู่ในรูปแบบของค่าความน่าจะเป็น ดังนั้นควรมีการจำแนกหมวดหมู่ให้ชัดเจนว่าหมวดหมู่ใดเป็นผลลัพธ์การทำนายที่แท้จริง โดยปกติแล้ว การ

จำแนกข้อมูลที่มีมากกว่าสองหมวดหมู่มักจะอยู่ในรูปแบบของการจำแนกแบบหลายคลาส (Multi-class classification) หลักการจำแนกแบบหลายคลาส คือ การเลือกผลการทำนายค่าความน่าจะเป็นที่มีสูงสุดเป็นคำตอบ ซึ่งมีเพียงคำตอบเดียวเท่านั้น อย่างไรก็ตาม เส้นทางบนกราฟสามารถมีทางแยกได้หลายเส้นทาง หากใช้การจำแนกแบบหลายคลาสจำแนกผลการทำนายเส้นทางบนกราฟปริภูมิสถานะทำให้ได้ผลคำตอบเพียงหนึ่งทางเท่านั้น ส่งผลให้บางเส้นทางอาจไม่ถูกทำนายออกมาเลย หากต้องการผลการทำนายเส้นทางทั้งหมดบนกราฟปริภูมิสถานะที่มีจะต้องมีผลการทำนายมากกว่าหนึ่งคำตอบ วิธีการจำแนกประเภทแบบหลายป้ายจึงเป็นวิธีที่เหมาะสมสำหรับการจำแนกหมวดหมู่เส้นทางทั้งหมดบนกราฟปริภูมิสถานะ

### 2.1.2.2 การจำแนกแบบหลายป้าย (Multi-label classification)

การจำแนกแบบหลายป้าย (Multi-label classification) [34] เป็นการจำแนกที่มีลักษณะทั่วไปของการจำแนกหลายคลาส แต่แตกต่างกันที่ผลการทำนายสามารถทำนายคำตอบได้มากกว่าหนึ่งคำตอบ เช่น การจำแนกส่วนประกอบที่อยู่บนรูปภาพผลไม้สีเขียว การทำนายแบบการจำแนกแบบหลายป้ายสามารถมีรูปอะโวคาโด มะม่วงและฝรั่งได้ ในขณะที่การจำแนกหลายคลาสอาจจะมีคำตอบเป็น อะโวคาโด มะม่วง หรือฝรั่งอย่างใดอย่างหนึ่งเท่านั้น การจำแนกแบบหลายป้ายถูกใช้กันอย่างแพร่หลาย เช่น การจัดประเภทข้อความ การจัดประเภทรูปภาพและการจัดประเภทของหนังสือ ที่นิยมใช้ในการเลือกผลการทำนายของการจำแนกหลายป้าย คือ กลยุทธ์การหาขีดแบ่ง โดยการกำหนดค่าคงที่หนึ่งตัวเป็นขีดแบ่ง เพื่อเป็นตัวแบ่งผลลัพธ์การทำนายที่อยู่ในรูปแบบของค่าความน่าจะเป็น เช่น กำหนดขีดแบ่งเท่ากับ 0.7 หากค่าความน่าจะเป็นของหมวดหมู่ใดมีค่ามากกว่าหรือเท่ากับ 0.7 หมวดหมู่นั้นจะเป็นผลลัพธ์ของการทำนาย แต่หากมีค่าน้อยกว่าจะหมายความว่าหมวดหมู่นั้นไม่ใช่ผลการทำนาย

ก่อนจะกล่าวถึง ตัววัดการประเมินผลของการจำแนกหลายป้าย (Evaluation metric for Multi-label classification) สิ่งหนึ่งที่สำคัญ คือ การเข้าใจตัวแปรที่อ้างในตัววัดการประเมิน มีดังนี้

- 1) ผลบวกจริง (True positive) คือ ผลการทำนายตรงกับผลที่เกิดขึ้นจริง ในกรณีที่ทำนายว่าเป็นจริงและผลที่เกิดขึ้นจริงก็เป็นจริงเช่นเดียวกัน
- 2) ผลลบจริง (True negative) คือ ผลการทำนายตรงกับผลที่เกิดขึ้นจริง ในกรณีที่ทำนายว่าเป็นเท็จและผลที่เกิดขึ้นจริงก็เป็นเท็จเช่นเดียวกัน
- 3) ผลบวกปลอม (False positive) คือ ผลการทำนายไม่ตรงกับผลที่เกิดขึ้นจริง โดยที่ผลการทำนายว่าเป็นจริง แต่ผลที่เกิดขึ้นจริงเป็นเท็จ
- 4) ผลลบปลอม (False negative) คือ ผลการทำนายไม่ตรงกับผลที่เกิดขึ้นจริง โดยที่ผลการทำนายว่าเป็นเท็จ แต่ผลที่เกิดขึ้นจริงเป็นจริง

การประเมินประสิทธิภาพของการจำแนกหลายป้าย ต้องพิจารณาจากการทำนายที่อาจถูกต้องบางส่วน ถูกต้องทั้งหมดหรือไม่ถูกต้องเลย ดังนั้น ตัววัดสำหรับการจำแนกหลายป้ายที่ประยุกต์การกำหนดขีดแบ่งที่เหมาะสมร่วมด้วย [35, 36] มีดังนี้

- 1) ค่าระยะที่หายไป (Hamming loss หรือ HL) คือ การคำนวณจากผลการทำนายของป้ายที่จำแนกผิด ซึ่งค่ายิ่งน้อยจะแสดงให้เห็นว่าผลการทำนายนั้นมีประสิทธิภาพ
- 2) ค่าความเที่ยงระดับไมโคร (Micro-Precision หรือ MP) คือ สัดส่วนผลรวมของผลบวกจริงของการจำแนกป้ายใดๆ ต่อผลรวมของผลบวกจริงและผลรวมของผลบวกปลอมของการจำแนกป้ายนั้น
- 3) ค่าเรียกคืนระดับไมโคร (Micro-Recall หรือ MR) คือ สัดส่วนผลรวมของผลบวกที่แท้จริงของการจำแนกป้ายใดๆ ต่อผลรวมของผลบวกจริงและผลรวมของผลบวกปลอมของการจำแนกป้ายนั้น
- 4) เอฟวันระดับไมโคร (Micro-F1 หรือ MF1) คือ ค่าเฉลี่ยแบบค่าเฉลี่ยฮาร์โมนิกของค่าความเที่ยงระดับไมโครและค่าเรียกคืนระดับไมโคร

### 2.1.2.3 กลยุทธ์การหาขีดแบ่ง (Threshold strategy)

กลยุทธ์การหาขีดแบ่ง (Threshold strategy) คือ กระบวนการในการหาค่าคงที่ให้เป็นขีดแบ่ง (Threshold) เพื่อเลือกผลการทำนายที่แท้จริง ซึ่งกลยุทธ์การหาขีดแบ่งถูกใช้ในการจัดการกับปัญหาการจำแนกแบบหลายป้าย [16] โดยกำหนดค่าคงที่หนึ่งตัวเพื่อเป็นขีดแบ่งที่เหมาะสม โดยการทำงานเริ่มต้นเมื่อได้ผลลัพธ์การทำนายที่อยู่ในรูปแบบของค่าความน่าจะเป็น จากนั้น พิจารณาแต่ละหมวดหมู่โดยการเปรียบเทียบกับขีดแบ่งที่กำหนด ค่าความน่าจะเป็นที่ถูกเปรียบเทียบกับขีดแบ่งจะถูกแปลงให้อยู่ในรูปแบบของเวกเตอร์ทวิภาค (Binary vector) เพื่อให้ง่ายต่อการแปลงเป็นผลคำตอบที่ชัดเจน โดยหมวดหมู่ที่มีผลการทำนายค่าความน่าจะเป็นมากกว่าหรือเท่ากับขีดแบ่งจะถูกแปลงเป็นเลข 1 หมายความว่าหมวดหมู่นั้นเป็นผลลัพธ์การทำนาย แต่หากมีค่าน้อยกว่าขีดแบ่งจะถูกแปลงเป็นเลข 0 หมายความว่าหมวดหมู่นั้นไม่ใช่ผลลัพธ์การทำนาย

อย่างไรก็ตาม การกำหนดขีดแบ่งที่เหมาะสมกับข้อมูลจริงไม่ใช่เรื่องง่าย เพราะหากกำหนดขีดแบ่งที่ไม่เหมาะสมกับข้อมูลเหล่านั้น มักส่งผลกระทบต่อความถูกต้องของการทำนายของโครงข่ายประสาทเทียม โดยทั่วไป มีการกำหนดขีดแบ่งไว้หลายแบบ เช่น การใช้ค่าโดยปริยายเท่ากับ 0.5 ซึ่งเป็นการกำหนดที่ง่ายและถูกใช้ในหลายงานวิจัย [13, 14] แต่ขีดแบ่งโดยค่าปริยายอาจไม่ได้แสดงถึงการแบ่งที่เหมาะสมที่สุดและอาจส่งผลให้ผลการทำนายมีประสิทธิภาพต่ำ อย่างไรก็ตาม มีหลายงานวิจัยที่ได้นำเสนอวิธีการกำหนดขีดแบ่งด้วยวิธีต่างๆ เช่น งานวิจัย [37] นำเสนอวิธีการเลือกค่าสูงที่สุด (Maximum cut) ของค่าที่ทำนาย งานวิจัย [38] นำเสนอวิธีใช้คะแนนเพื่อปรับขีดแบ่งที่ได้จาก



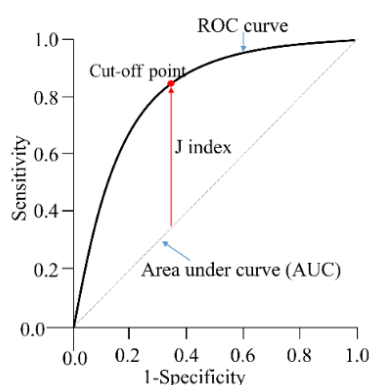
ตัวทำนายของแต่ละคู่ของหมวดหมู่ (Score-based method) และวิธีการจัดลำดับเพื่อปรับขีดแบ่ง (Rank-based method) แต่วิธีที่กล่าวมาข้างต้นไม่ได้พิจารณาการทำงานจริงเมื่อมีการกำหนดขีดแบ่ง นอกจากนั้น มีหลายงานวิจัย [8, 10, 11, 15] นำเสนอการจำแนกแบบหลายป้ายและได้เลือกใช้ฟังก์ชันซอฟต์แวร์แมกซ์ (Softmax function) ซึ่งถูกใช้งานบ่อยในงานจำแนกประเภทหมวดหมู่ เพื่อเรียงลำดับจากค่าที่มากที่สุดไปยังค่าที่น้อยที่สุด แต่ฟังก์ชันซอฟต์แวร์แมกซ์มักเกิดปัญหาด้านประสิทธิภาพการทำงาน เมื่อข้อมูลที่ทำนายมีปริมาณหมวดหมู่เป็นจำนวนมาก ซึ่งการใช้ฟังก์ชันซอฟต์แวร์แมกซ์เหมาะกับการทำนายที่คาดหวังผลลัพธ์ที่ถูกต้องเพียงค่าเดียวเท่านั้น

ดังนั้น ในงานวิจัยนี้ได้เลือกใช้กลยุทธ์การหาขีดแบ่ง ซึ่งเป็นการกระบวนกรหาขีดแบ่งที่เหมาะสมกับข้อมูลในงานวิจัย โดยได้นำทฤษฎีของ ยูเดน เจ-อินเด็กซ์ (Youden's J-Index) หรือเรียกว่า เจ-อินเด็กซ์ (J-index) [39] มาประยุกต์ใช้ โดยค่า J-index แสดงตำแหน่งบนพื้นที่ได้เส้นโค้งบนกราฟหรือเส้นโค้ง ROC (Receiver operating characteristic curve หรือ ROC curve) [40] และค่า J-index ที่ค่ามากที่สุดจะถือว่าเป็นจุดตัด (Cut-off point) ที่ใช้สำหรับเป็นเกณฑ์ที่ดีที่สุดของแต่ละหมวดหมู่ ซึ่งพิจารณาจาก ความไว (Sensitivity) จากอัตราบวกจริง (True positive rate) และความจำเพาะ (Specificity) จากอัตราบวกเท็จ (False positive rate) บนหน่วยเมตริกบนเส้นโค้ง ROC แสดงในรูปที่ 2.10 สามารถเขียนเป็นสมการได้ ดังต่อไปนี้

$$J = \text{sensitivity} + \text{specificity} - 1$$

$$\text{sensitivity} = \frac{tp_s}{tp_s + fn_s} \quad (1)$$

$$\text{specificity} = \frac{tn_s}{tn_s + fp_s}$$



รูปที่ 2.10 ตัวอย่างจุดตัดของเกณฑ์ที่ดีที่สุดโดยใช้ทฤษฎีของ J-index บนเส้นโค้ง ROC [40]

เมื่อได้เกณฑ์ที่ดีที่สุดของแต่ละหมวดหมู่แล้ว จากนั้น ผู้วิจัยได้นำเกณฑ์ที่ดีที่สุดของแต่ละหมวดหมู่มาพิจารณา โดยเลือกค่าของเกณฑ์ที่มีค่าที่น้อยที่สุดออกมาและกำหนดให้เป็นขีดแบ่งที่เหมาะสมเพียงหนึ่งขีดแบ่ง ซึ่งขีดแบ่งที่กำหนดจะถูกนำไปใช้ในการจำแนกสำหรับระบบที่สนใจ โดยนำมาเปรียบเทียบว่าหมวดหมู่ใดควรได้รับการจำแนกให้เป็นผลการทำนายที่แท้จริง ซึ่งทำให้ได้ผลการทำนายที่ถูกต้องและครบถ้วน อีกทั้งยังช่วยเพิ่มประสิทธิภาพของการทำนาย โดยวิธีการพิจารณาขีดแบ่งที่เหมาะสม จะกล่าวในลำดับถัดไป

## 2.2 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่มีความเกี่ยวข้องกับงานวิจัยนี้ สามารถแบ่งเป็น 3 ส่วน ส่วนแรกแรกสำรวจงานวิจัยที่ใช้วิธีการแบบดั้งเดิมในการแก้ไขปัญหาการระเบิดของปริภูมิสถานะ ส่วนที่สองสำรวจงานวิจัยที่มีการประยุกต์ใช้อัลกอริทึมการเรียนรู้ของเครื่องในโมเดลเชิงกึ่ง โดยเน้นไปที่การใช้ อัลกอริทึมการเรียนรู้ของเครื่องเพื่อจัดการกับปัญหาการระเบิดของปริภูมิสถานะโดยเฉพาะและส่วนสุดท้าย สำรวจงานวิจัยที่ศึกษาวิธีการนำแบบจำลองแอลเอสทีเอ็มและการจำแนกแบบหลายป้าย ไปใช้กับโดเมนต่างๆ

### 2.2.1 งานวิจัยที่นำวิธีการแบบดั้งเดิมในการแก้ไขปัญหาการระเบิดของปริภูมิสถานะ

จากการสำรวจ ผู้วิจัยพบว่ามียางานวิจัยที่สนใจการจัดการกับการระเบิดของปริภูมิสถานะในโมเดลเชิงกึ่ง โดยใช้วิธีการแบบดั้งเดิม หนึ่งในวิธีการพื้นฐาน คือการลดสถานะ (State reduction) ซึ่งมีจุดประสงค์เพื่อลดจำนวนสถานะที่ซ้ำซ้อนกันในปริภูมิสถานะ

งานวิจัย[41] นำเสนออัลกอริทึมที่เพิ่มประสิทธิภาพการค้นหา (Search-based optimization) เพื่อค้นหาสถานะในปริภูมิสถานะของระบบพร้อมกัน ซึ่งการเพิ่มประสิทธิภาพนี้จะช่วยหลีกเลี่ยงการระเบิดของปริภูมิสถานะได้ นอกจากนี้ งานวิจัยนี้สนใจการทวนสอบเพื่อหาการหยุดชะงัก (Deadlocks) โดยใช้เทคนิคการตัดแต่งกิ่งปริภูมิสถานะ (State space pruning) งานวิจัย [42] ใช้จินเนติกอัลกอริทึม (Genetic algorithms) สำหรับสำรวจปริภูมิสถานะที่มีขนาดใหญ่ของระบบพร้อมกัน เพื่อค้นหาสถานะที่ผิดพลาด อย่างไรก็ตาม งานวิจัยนี้สามารถตรวจจับการหยุดชะงักและการละเมิดการยืนยัน (Assertion violations) ได้เท่านั้น ในงานวิจัย [43] ได้นำเสนออัลกอริทึมของการลดความสมมาตร (Symmetry reduction) เพื่อลดขนาดของปริภูมิสถานะ อย่างไรก็ตาม อัลกอริทึมนี้จะถูกใช้เพื่อการหาจุดบกพร่องของระบบเป็นหลัก ซึ่งอาจจะทำให้เกิดผลบวกวงขึ้นได้แทนที่จะสำรวจสถานะของระบบ งานวิจัย [44] เสนอกลยุทธ์การจัดเก็บ (Storing strategy) โดยใช้วิธีการแบบสุ่ม เพื่อลดจำนวนสถานะที่เข้าถึงในระหว่างการสำรวจปริภูมิสถานะ อย่างไรก็ตาม งานวิจัยนี้ใช้ได้กับระบบที่กำหนดเวลาไว้เท่านั้นและวิธีการขนาดปริภูมิสถานะในการทวนสอบ

(Partial order reduction) ยังอยู่ในระหว่างการพัฒนา งานวิจัย [45] เสนอวิธีการใช้ประโยชน์จากการลดความสมมาตรเพื่อแก้ไขปัญหาการระเบิดของปริภูมิสถานะ ในการทวนสอบแบบจำลอง แต่วิธีการของพวกเขาเน้นเฉพาะเจาะจงกับการลดความสมมาตรในปี แมชชีน (B machine) และจำเป็นต้องปรับปรุงเวลาในการดำเนินการ ในงานวิจัย [46] ได้เสนออัลกอริทึมแบบอาณานิคมมด (Ant colony optimization) เพื่อค้นหาข้อผิดพลาดที่ใกล้เคียงที่สุด โดยใช้เทคนิคสุ่มความน่าจะเป็น (Stochastic technique) เพื่อตรวจสอบคุณสมบัติด้านความปลอดภัยในระบบที่ทำงานพร้อมกันขนาดใหญ่ นอกจากนี้ อัลกอริทึมแบบอาณานิคมมดยังใช้เพื่อตรวจสอบคุณสมบัติความดำเนินชีวิตในระบบพร้อมกัน ในงานวิจัย [47] ซึ่งเป็นการตรวจจับเส้นทางที่สั้นที่สุดเพื่อค้นหาการหยุดชะงักในระบบที่เกิดขึ้นพร้อมกัน

จากงานวิจัยกล่าวข้างต้น เป็นเพียงการนำเสนอวิธีการเสนออัลกอริทึมเพื่อลดขนาดปริภูมิสถานะให้มีขนาดเล็กลงหรือใช้วิธีการตัดทอนบางสถานะที่ไม่สนใจ เพื่อให้ประหยัดเวลาในการสำรวจปริภูมิสถานะและหน่วยความจำในการสร้างปริภูมิสถานะและงานวิจัยที่กล่าวถึงนั้น ไม่ได้ครอบคลุมถึงการนำสูตรแอลทีแอลมาใช้ในการทวนสอบระบบในโมเดลเช็กกิง

## 2.2.2 งานวิจัยที่ประยุกต์ใช้แมชชีนเลิร์นนิงในโมเดลเช็กกิง

เมื่อได้ค้นคว้างานวิจัยที่นำเสนอวิธีการแบบดั้งเดิมแล้ว ผู้วิจัยได้ไปค้นคว้างานวิจัยที่นำแมชชีนเลิร์นนิงมาประยุกต์ใช้ในโมเดลเช็กกิงเพื่อจัดการกับปัญหาการระเบิดของปริภูมิสถานะ

งานวิจัย [48] นำวิธีการแบบใหม่โดยนำการเรียนรู้แบบเสริมกำลัง (Reinforcement learning) มาใช้ในการติดตามลำดับของสถานะในปริภูมิสถานะ เพื่อทวนสอบคุณสมบัติความดำเนินชีวิตในระบบแบบพร้อมกัน แต่วิธีการนี้จะทำให้เกิดเส้นทางแบบวนรอบไม่รู้จบ ในทำนองเดียวกัน งานวิจัย [49] ใช้วิธีการแบบเสริมกำลัง เพื่อเพิ่มประสิทธิภาพในการทำออน-เดอะ-ฟลายสำหรับระบบที่มีขนาดใหญ่ในโมเดลเช็กกิง ซึ่งสามารถได้ผลลัพธ์เป็นการประมาณเท่านั้น ถึงแม้ว่างานวิจัยนี้สามารถจัดการกับระบบที่มีขนาดใหญ่ได้ แต่วิธีการนี้ก็ยังสามารถทำงานได้ดีในระบบที่มีขนาดเล็ก นอกจากนี้ การประเมินประสิทธิภาพเป็นเพียงการวัดหน่วยความจำเป็นหลัก ซึ่งไม่ได้วัดความแม่นยำของวิธีการ งานวิจัย [50] ได้นำแมชชีนเลิร์นนิงโดยใช้วิธีทางสถิติแบบเกาส์เซียน (Gaussian statistical approaches) เพื่อทวนสอบพารามิเตอร์ทางสถิติในโมเดลเช็กกิง อย่างไรก็ตาม งานวิจัยนี้ไม่ได้แก้ปัญหการระเบิดของปริภูมิสถานะในโมเดลเช็กกิง งานวิจัย [51] นำแมชชีนเลิร์นนิงมาทำนายคุณสมบัติชั่วคราว (Temporal properties) ของระบบส่งสัญญาณบนกราฟ โดยวิธีการนี้เป็นการทำนายคุณสมบัติชั่วคราวเท่านั้น ซึ่งไม่จำเป็นต้องค้นหาสถานะทั้งหมดของปริภูมิสถานะ งานวิจัย [52] นำเสนอแนวทางของการสำรวจปริภูมิสถานะเพื่อจัดการปริภูมิสถานะสำหรับการตรวจสอบการเข้าถึงและจัดการกับปัญหาการระเบิดของปริภูมิสถานะของระบบซอฟต์แวร์ที่ซับซ้อน โดยใช้เทคนิค

ของแมชชีนเลิร์นนิงโดยวิธีการรวมกลุ่ม (Ensemble methods) แบบวิธีการบูสต์ติง (Boosting method) โดยเป็นการนำข้อผิดพลาดของแบบจำลองแมชชีนเลิร์นนิงครั้งก่อนมาใช้ฝึกแบบจำลองแมชชีนเลิร์นนิงใหม่ ซึ่งทุกแบบจำลองแมชชีนเลิร์นนิงจะใช้ข้อมูลการเรียนรู้ชุดเดียวกัน แนวทางของงานวิจัยนี้มีความแม่นยำสำหรับการทำนายเส้นทางที่มีคาดหวังและสามารถค้นหาสถานะเป้าหมายได้รวดเร็วเมื่อเทียบกับแบบจำลองพื้นฐานอื่น แต่แนวทางที่งานวิจัยนี้นำเสนอมีข้อจำกัด คือ การสร้างแบบจำลองที่มีขนาดเล็กเป็นงานที่ซับซ้อน ซึ่งบางกรณีไม่สามารถสร้างได้ นอกจากนี้ ปริมาณและคุณภาพของแบบจำลองขนาดเล็กจะส่งผลถึงประสิทธิภาพการทำงานของแนวทางที่งานวิจัยนี้นำเสนอ นอกจากนี้ งานวิจัย [53] ได้นำเสนอวิธีการโครงข่ายประสาทเทียมแบบกราฟ (Graph neural network) เพื่อทำการทวนสอบแอลที่แอลในโมเดลเชิงกึ่งกับระบบที่โมเดลผ่านแบบจำลองของบูชชีโอโตมาตา ซึ่งวิธีการนี้เหมาะกับแบบจำลองที่มีขนาดเล็กถึงปานกลางเท่านั้น ในขณะที่งานวิจัย [54] นำเสนอการเรียนรู้แบบเสริมกำลังเชิงลึก เพื่อค้นหาคุณสมบัติการเข้าถึงสถานะได้ในระบบแบบเฉพาะ อย่างไรก็ตาม งานวิจัยนี้อาจจะทำให้เกิดการระเบิดของปริภูมิสถานะเกิดขึ้นได้ เมื่อนำวิธีการนี้ไปใช้ในระบบที่มีความซับซ้อน

นอกจากนี้พบงานวิจัยที่มีความน่าสนใจคือ งานวิจัย [55] ที่ได้นำเสนออัลกอริทึมการจำแนกประเภทไบนารีในแมชชีนเลิร์นนิงที่มีอยู่ คือ บูตทรีส์ (Boosted tree) แรนดอมฟอเรส (Random forest) ต้นไม้การตัดสินใจ (Decision tree) และการถดถอยโลจิสติกส์ (Logistic regression) โดยมีวัตถุประสงค์คือหลีกเลี่ยงการระเบิดของปริภูมิสถานะในแบบจำลองที่โมเดลโดยใช้โครงสร้างคริปกี (Kripke structure) [56] และสูตรแอลที่แอลที่ถูกสร้างขึ้นแบบสุ่ม อัลกอริทึมนี้ได้เรียนรู้โครงสร้างคริปกีและสูตรแอลที่แอลและผลลัพธ์ของสูตรแอลที่แอล ผลการทดลองแสดงให้เห็นว่าอัลกอริทึมที่มีอยู่มีความแม่นยำเท่ากับการทวนสอบแบบจำลองและสูตรแอลที่แอลที่เคยเรียนรู้ อย่างไรก็ตาม ความยาวที่แตกต่างกันของสูตรแอลที่แอล มีความเฉพาะเจาะจงกับชุดข้อมูลและหากไม่ได้นำบางแอลที่แอลไปให้อัลกอริทึมเรียนรู้ ก็อาจจะไม่สามารถให้ผลลัพธ์ที่ถูกต้องได้ ในทำนองเดียวกัน ในงานวิจัย [57] ได้นำเสนออัลกอริทึมการจำแนกประเภทไบนารีในแมชชีนเลิร์นนิงที่มีอยู่และใช้แบบจำลองคริปกีเช่นเดียวกับงานวิจัย [55] แต่งานวิจัยนี้จะมุ่งเน้นการทวนสอบโดยใช้สูตรแอลที่แอล อย่างไรก็ตาม ตัวอย่างที่งานวิจัยนี้ใช้ทดสอบเป็นการสุ่มสร้างโครงสร้างคริปกีและสูตรแอลที่แอลเท่านั้น โดยไม่ได้ใช้โครงสร้างคริปกีที่จำลองการทำงานของระบบจริงและประสิทธิภาพสำหรับการทำนายผลลัพธ์การทวนสอบจะขึ้นอยู่กับข้อมูลสำหรับการเรียนรู้ ถ้าหากเตรียมข้อมูลไม่ครอบคลุมก็อาจจะทำให้ได้ผลลัพธ์การทวนสอบที่ผิดพลาดได้เช่นเดียวกับงานวิจัยก่อนหน้า จากการค้นคว้างานวิจัยสองฉบับนี้ ผู้วิจัยได้เห็นประโยชน์ของการนำแมชชีนเลิร์นนิงอัลกอริทึมมาเรียนรู้ข้อมูลสำหรับการทวนสอบระบบ เพื่อให้ได้วิธีการใหม่ที่จะช่วยหลีกเลี่ยงการระเบิดของปริภูมิสถานะ แต่งานวิจัยได้ใช้วิธีการสุ่มสร้างโครงสร้างคริปกีที่ไม่ได้เป็นการจำลองการทำงานของระบบจริงและไม่ได้

ใช้วิธีการค้นหาปริภูมิสถานะ อย่างไรก็ตาม งานวิจัยนี้สนใจโมเดลเชิงกึ่งเช่นเดียวกับงานวิจัยนี้ แต่มีการเลือกใช้ตรรกศาสตร์เวลาเชิงเส้นที่แตกต่างกัน โดยใช้สูตรแอลทีแอล นอกจากนี้ วิธีการทวนสอบมีความแตกต่างกันโดยสิ้นเชิงและใช้วิธีการค้นหาภายในปริภูมิสถานะแบบออน-เดอะ-ฟลาย เพื่อทำการทวนสอบการทำงานของระบบ ส่วนงานวิจัยนี้ใช้วิธีการทำนายผลการทวนสอบการทำงานของระบบโดยใช้แมชชีนเลิร์นนิงอัลกอริทึมที่ผ่านการเรียนรู้

งานวิจัยที่ได้กล่าวข้างต้น เป็นเพียงการนำเสนอวิธีการของแมชชีนเลิร์นนิงที่มีอยู่มาประยุกต์ใช้ในโมเดลเชิงกึ่ง ซึ่งไม่ได้เป็นวิธีการแบบใหม่และยังไม่มีงานวิจัยใดเลยที่นำอัลกอริทึมของแมชชีนเลิร์นนิงที่สนใจไปใช้ในการทำนายเส้นทางบนปริภูมิสถานะและทำนายผลการทวนสอบคุณสมบัติต่างๆ ของระบบ เช่น ความปลอดภัยและความดำเนินชีวิต ของแบบจำลองเพทรีเน็ตโดยใช้สูตรแอลทีแอล

### 2.2.3 งานวิจัยที่นำแบบจำลองแอลเอสทีเอ็มและการจำแนกแบบหลายป้าย

ในช่วงไม่กี่ปีที่ผ่านมา แบบจำลองแอลเอสทีเอ็มได้ถูกกล่าวถึงในวงกว้างและมีประสิทธิภาพในการจัดการปัญหาการจำแนกแบบหลายป้ายและจัดการกับปัญหาข้อมูลที่มีความยาวได้อย่างมีประสิทธิภาพ มีงานวิจัยหลากหลายสาขาได้นำแบบจำลองแอลเอสทีเอ็มไปประยุกต์ใช้

ในสาขาทางการแพทย์ งานวิจัย [9] แอลเอสทีเอ็มถูกใช้เป็นตัวถอดรหัสเพื่อใช้ในการสร้างเครื่องแปลยาจีนโบราณที่สามารถแปลอาการป่วยไปยังยาสมุนไพรจีนที่สอดคล้องกับอาการ เครื่องแปลยาจีนที่ถูกพัฒนาขึ้นมีประสิทธิภาพในการทำนายยาสมุนไพรได้ดีกว่าแบบจำลองพื้นฐานอื่นๆ แต่เมื่อนำผลการทำนายมาเปรียบเทียบกับผลการทำนายโดยผู้เชี่ยวชาญพบว่าเครื่องแปลยาจีนยังทำนายได้ไม่ถูกต้องเท่าไรนัก และงานวิจัย [12] นำเสนอแบบจำลองแอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้าย เพื่อทำการวินิจฉัยผู้ป่วยขั้นวิกฤต โดยการเรียนรู้ข้อมูลทางการแพทย์จากคลินิก ซึ่งแบบจำลองของงานวิจัยนี้ประสบความสำเร็จในการจำแนกการวินิจฉัยผู้ป่วยวิกฤต โดยอ้างอิงจากข้อมูลทางการแพทย์จากคลินิก

ในสาขากระบวนการภาษาธรรมชาติ (Natural language processing) งานวิจัย [8] ได้นำเสนอแบบจำลองการสร้างลำดับของการจำแนกแบบหลายป้าย โดยเสนอโครงสร้างที่มีตัวถอดรหัสแบบใหม่ที่สนใจความสัมพันธ์ระหว่างป้าย เพื่อแก้ปัญหาการสร้างลำดับของการจำแนกแบบหลายป้าย วิธีการของงานวิจัยนี้ได้แนวคิดของวิธีการแบบลำดับต่อลำดับ ซึ่งแอลเอสทีเอ็มถูกใช้เป็นตัวเข้ารหัสและตัวถอดรหัส เพื่อหาค่าที่คาดว่าจะเป็ผลลัพท์มากที่สุด ในงานวิจัยนี้พิจารณาความสัมพันธ์ระหว่างป้าย ซึ่งอาจมีส่วนที่ช่วยให้แบบจำลองทำนายได้ถูกต้องมากยิ่งขึ้น แต่เมื่อวัดประสิทธิภาพของการทำนายในงานวิจัยนี้พบว่า ประสิทธิภาพด้านความแม่นยำยังไม่ดีเท่าไรนัก งานวิจัย [7] นำเสนอแบบจำลองลำดับต่อลำดับแบบใหม่สำหรับการจำแนกประเภทข้อความแบบ

หลายป้าย โดยพิจารณาความสัมพันธ์ระหว่างป้าย โดยแอลเอสทีเอ็มถูกใช้เป็นตัวถอดรหัส เพื่อใช้ในการทำนายการจำแนกแบบหลายป้าย โดยใช้ฟังก์ชันซอฟต์แวร์แม็กซ์เพื่อหาผลลัพธ์ที่เป็นไปได้มากที่สุด จากผลการทดลองแบบจำลองของงานวิจัยนี้ยังมีประสิทธิภาพที่ไม่ดีเท่าไรนัก โดยมีค่าความถูกต้องและค่าความแม่นยำน้อยกว่า 90% ในงานวิจัย [10] ได้นำเสนอแบบจำลองแบบใหม่โดยการนำโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional neural network) มารวมกับโครงข่ายประสาทเทียมแบบวนซ้ำ (Recurrent neural network) สำหรับการทำนายแบบหลายป้าย อย่างไรก็ตาม งานวิจัยนี้ใช้ชุดข้อมูลที่มีเพียงหนึ่งหมวดหมู่เท่านั้นและโมเดลของพวกเขาใช้ฟังก์ชันซอฟต์แวร์แม็กซ์ ซึ่งประสิทธิภาพของแบบจำลองพื้นฐานทั้งหมดดีกว่าแบบจำลองของพวกเขาและวิธีการนี้อาจนำไปสู่ปัญหา โอเวอร์ฟิตติ้ง (Overfitting) ได้ ในทำนองเดียวกัน งานวิจัย [11] ได้นำแอลเอสทีเอ็มแบบสองทิศทางไปใช้ในการจำแนกแบบหลายขั้นตอน (Multi-step classification) เพื่อเพิ่มความแม่นยำและลดผลการทำนายที่ผิดพลาด แม้ว่าการวิจัยนี้จะมีประสิทธิภาพที่ดีกว่าแบบจำลองพื้นฐานที่มีอยู่ แต่การวิจัยนี้ไม่ได้ใช้ความแม่นยำระดับไมโครในการวัดแบบจำลองของพวกเขาและใช้ฟังก์ชันซอฟต์แวร์แม็กซ์เพื่อเลือกคะแนนสูงสุดเป็นผลลัพธ์การทำนาย

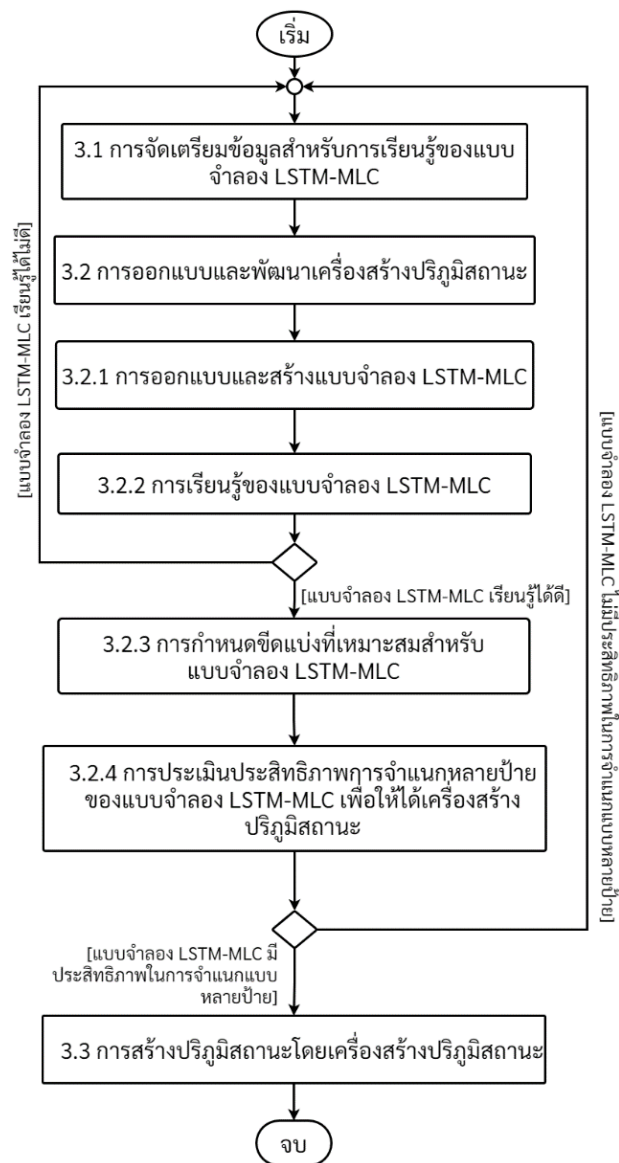
จากการค้นคว้ามีการศึกษาเพียงไม่กี่ชิ้นเท่านั้นที่ได้นำแอลเอสทีเอ็มมาใช้ในกระบวนการทวนสอบ ซึ่งในงานวิจัย [58] ได้นำเสนอแบบจำลองแบบใหม่โดยการใช้แอลเอสทีเอ็ม เพื่อใช้ในการตรวจสอบปัญหาการเลือกอัลกอริทึมในการทวนสอบซอฟต์แวร์ อย่างไรก็ตาม ในงานวิจัยนี้ไม่ได้พิจารณาปัญหาการระเบิดของปริภูมิสถานะ

จากการค้นคว้างานวิจัยทั้งหมดที่เกี่ยวข้อง มีหลายงานวิจัยที่นำแมชชีนเลิร์นนิงมาใช้ในหลากหลายสาขาซึ่งรวมถึงการทำโมเดลเชิงกึ่ง แต่อย่างไรก็ตาม ยังไม่มีการศึกษาใดที่ใช้แบบจำลองแอลเอสทีเอ็มมาประยุกต์ใช้ในการทำโมเดลเชิงกึ่ง นอกจากนี้วิธีการของแบบจำลองแอลเอสทีเอ็มได้นำไปประยุกต์ใช้กับปัญหาของการจำแนกแบบหลายป้าย ซึ่งมีการกำหนดขีดแบ่งโดยใช้วิธีการที่มีอยู่สำหรับการเลือกผลการทำนายที่แท้จริง จึงเป็นเรื่องที่น่าสนใจที่จะนำแนวทางของแบบจำลองแอลเอสทีเอ็มและการจำแนกแบบหลายป้ายมาใช้ในการบรรเทาปัญหาการระเบิดของปริภูมิสถานะในโมเดลเชิงกึ่งและใช้ในการทวนสอบคุณสมบัติของระบบ ดังนั้น ในงานวิจัยนี้มีจุดมุ่งหมายเพื่อนำเสนอกลยุทธ์การหาขีดแบ่งที่เหมาะสมเพื่อให้ได้ขีดแบ่งที่เหมาะสมกับลักษณะของข้อมูล ซึ่งจะถูกนำไปใช้ในเป็นส่วนหนึ่งของการพัฒนาเป็นเครื่องสร้างปริภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอลเอสทีเอ็มโดยใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้าย

### บทที่ 3

#### แนวคิดการพัฒนาเครื่องสร้างปริภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอลทีแอลโดยใช้แอลเอส ทีเอ็มที่มีการจำแนกแบบหลายป้าย

เมื่อได้ศึกษาและค้นคว้าความรู้ที่เกี่ยวข้องและงานวิจัยที่เกี่ยวข้องกับงานวิจัยนี้ ผู้วิจัยได้นำเสนอแนวคิดการพัฒนาเครื่องสร้างปริภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอลทีแอลโดยใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้าย ซึ่งกระบวนการนี้เป็นการสร้างปริภูมิสถานะเท่านั้น โดยที่ยังไม่มีส่วนของกระบวนการทวนสอบ แสดงแผนภาพขั้นตอนการพัฒนาเครื่องสร้างปริภูมิสถานะในรูปที่ 3.1

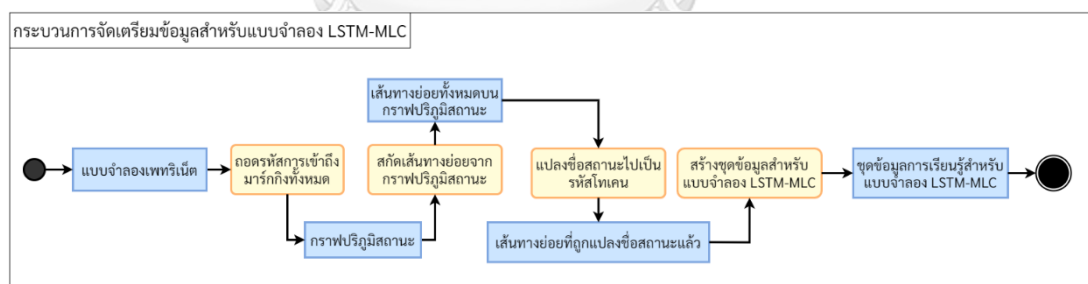


รูปที่ 3.1 แผนภาพขั้นตอนการพัฒนาเครื่องสร้างปริภูมิสถานะ

แนวคิดของงานวิจัยนี้เริ่มจากขั้นตอนการสร้างเครื่องสร้างปฏิภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอตที่แอลโดยใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้ายและหลังจากนั้นเป็นการสร้างปฏิภูมิสถานะและการทวนสอบคุณสมบัติการทำงานของระบบโดยการกำหนดสูตรแอตที่แอลแบบเฉพาะ โดยรายละเอียดของขั้นตอนการพัฒนาเครื่องสร้างปฏิภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอตที่แอลโดยใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้ายอธิบายในบทที่ 3 จากนั้น แนวคิดการพัฒนาเครื่องมือการทวนสอบคุณสมบัติการทำงานของระบบโดยการกำหนดสูตรแอตที่แอลแบบเฉพาะโดยใช้เครื่องสร้างปฏิภูมิสถานะ อธิบายในบทที่ 4

### 3.1 การจัดเตรียมข้อมูลสำหรับการเรียนรู้ของแบบจำลอง LSTM-MLC

ในขั้นตอนของการจัดเตรียมข้อมูลสำหรับการเรียนรู้ของแบบจำลองแอตที่เอ็มที่ถูกสร้างขึ้นในขั้นตอนถัดไป ขอเรียกว่า แบบจำลอง LSTM-MLC (LSTM-MLC model) มีขั้นตอนการเตรียมข้อมูลสำหรับการเรียนรู้ โดยใช้วิธีการจัดเตรียมข้อมูลทั่วไป [59] คือ เริ่มจากการทำความสะอาดข้อมูล (Data cleaning) การเลือกข้อมูล (Data selection) และการแปลงข้อมูล (Data transformation) เพื่อให้ได้ชุดข้อมูล (Dataset) ที่ทำให้แบบจำลอง LSTM-MLC สามารถเรียนรู้ได้อย่างมีประสิทธิภาพ โดยรับข้อมูลนำเข้าเป็นแบบจำลองเพทรีเน็ตมาจำลองบนเครื่องมือที่น่าและนำมาผ่านกระบวนการจัดเตรียมข้อมูล เพื่อให้ได้ผลลัพธ์เป็นชุดข้อมูลสำหรับการเรียนรู้สำหรับแบบจำลอง LSTM-MLC สามารถแสดงเป็นแผนภาพกิจกรรมได้ในรูปที่ 3.2



รูปที่ 3.2 แผนภาพกิจกรรมการเตรียมข้อมูลสำหรับการเรียนรู้ของแบบจำลอง LSTM-MLC

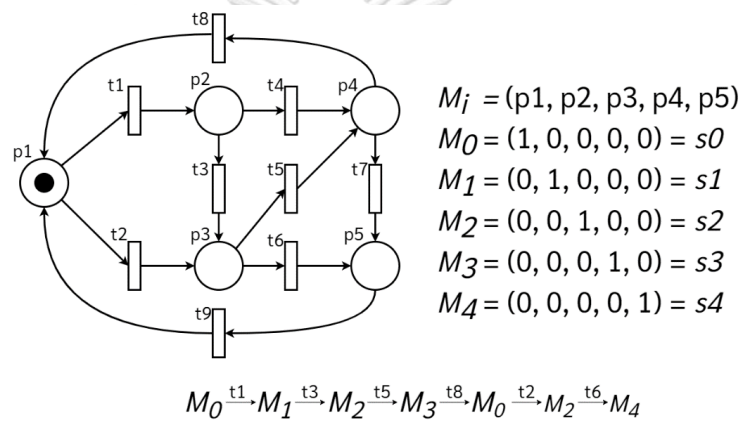
#### 3.1.1 การถอดรหัสการเข้าถึงมาร์กิ้งทั้งหมด

กระบวนการแรก คือ การทำความสะอาดข้อมูล โดยเริ่มจากการนำแบบจำลองเพทรีเน็ตของระบบที่เราสนใจมาหาสถานะการณืที่เป็นไปได้ทั้งหมด เช่น ระบบเครื่องรับจ่ายเงินอัตโนมัติ ระบบการสัญญาณไฟจราจรและระบบการส่งสินค้า เป็นต้น ซึ่งเหตุการณ์ที่เป็นไปได้ทั้งหมดในระบบถูกค้นหาด้วยวิธีการหาการเข้าถึงมาร์กิ้งทั้งหมด (All reachability marking) โดยเริ่มจากมาร์กิ้งเริ่มต้น



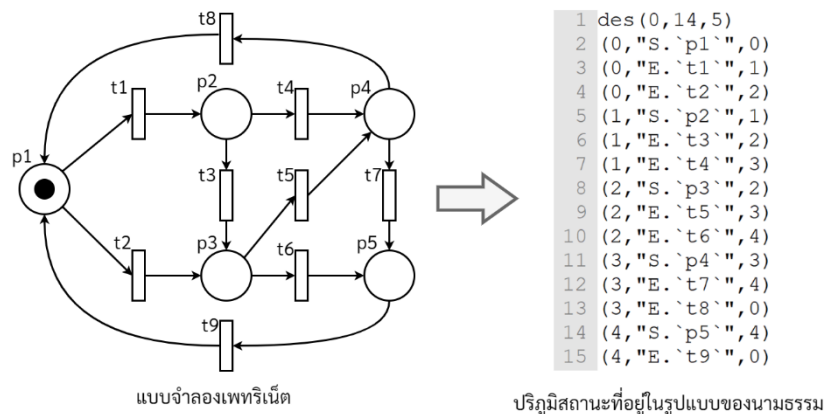
การเปิดใช้และการยิงของแต่ละทรานซิชัน ทำให้มาร์กกิงเปลี่ยนไป ซึ่งแต่ละมาร์กกิงแสดงจำนวนโทเคนที่อยู่ในเพลสเพื่อแสดงถึงสถานการณ์ที่เกิดขึ้นภายในระบบ

ตัวอย่างแบบจำลองเพทรีเน็ตที่ได้รับการแจกแจงมาร์กกิง แสดงตัวอย่างในรูปที่ 3.3 [22] โดยมาร์กกิงเริ่มต้นคือ  $M_0 = (1, 0, 0, 0, 0)$  แสดงให้เห็นว่าเพลส p1 มีโทเคน 1 อัน หมายถึงสถานะการเริ่มต้นทำงานของระบบ ดังนั้น สถานะเริ่มต้น  $s_0$  คือ มาร์กกิงเริ่มต้น  $M_0$  และจากมาร์กกิง  $M_0$  ไปสู่มาร์กกิงถัดไป ถ้าทรานซิชัน t1 ถูกเปิดใช้และยิงไปยังมาร์กกิงตัวถัดไป คือ มาร์กกิง  $M_1$  ซึ่งก็คือสถานะ  $s_1$  แต่ถ้า ทรานซิชัน t2 ถูกเปิดใช้และยิงไปยังมาร์กกิงตัวถัดไป คือ มาร์กกิง  $M_2$  ซึ่งก็คือสถานะ  $s_2$  สรุปได้ว่า มาร์กกิง  $M_i$  จะสามารถเขียนแทนด้วย สถานะ  $s_i$  ได้ ดังนั้น มาร์กกิงจะถูกเขียนใหม่ด้วยสถานะ  $s_0 s_1 s_2 s_3$  และ  $s_4$  ซึ่งสามารถแทนมาร์กกิง  $M_0 M_1 M_2 M_3$  และ  $M_4$  ตามลำดับ



รูปที่ 3.3 ตัวอย่างแบบจำลองเพทรีเน็ตและการแจกแจงมาร์กกิง

อย่างไรก็ดี การเข้าถึงมาร์กกิงทั้งหมด หมายถึง สถานการณ์ทั้งหมดที่สามารถเกิดขึ้นภายในระบบ โดยแต่ละมาร์กกิงแสดงให้เห็นว่ามีเพลสใดเป็นจริงบ้าง ซึ่งการค้นหาการเข้าถึงมาร์กกิงทั้งหมดเป็นวิธีมาตรฐานที่ใช้โดยทั่วไปและมีอยู่ในเครื่องมือโมเดลเช็กกิงที่ใช้วาดแบบจำลองเพทรีเน็ต เช่น เครื่องมือทีน่า เป็นต้น แต่อย่างไรก็ดี การเข้าถึงมาร์กกิงทั้งหมดมักถูกเข้ารหัสไว้ ทำให้ไม่สามารถนำไปใช้ได้โดยตรง ดังนั้นจึงต้องถอดรหัสการเข้าถึงมาร์กกิงที่ได้มาเพื่อให้สามารถนำไปใช้วาดกราฟปริภูมิสถานะของระบบ ตัวอย่างการเข้าถึงมาร์กกิงทั้งหมดที่ถูกเข้ารหัสในรูปแบบข้อความของแบบจำลองเพทรีเน็ต แสดงในรูปที่ 3.4



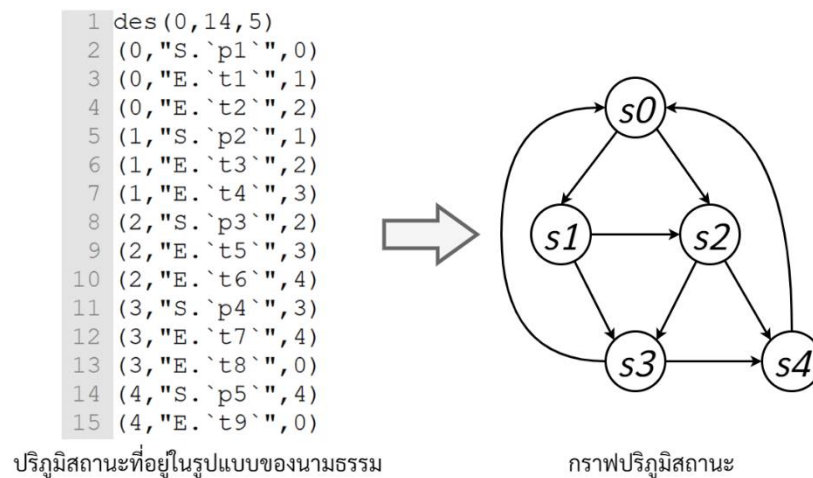
รูปที่ 3.4 ตัวอย่างการเข้าถึงมาร์กกิงทั้งหมดของแบบจำลองเพทรีเน็ต

ดังนั้น ผู้วิจัยจึงต้องนำการเข้าถึงมาร์กกิงทั้งหมดของแบบจำลองเพทรีเน็ตมาจัดให้อยู่ในรูปแบบของกราฟปริภูมิสถานะแทนต้นไม้เชิงการคำนวณ เพื่อป้องกันการแจกแจงสถานะแบบไม่สิ้นสุดสำหรับระบบที่มีการทำงานแบบวนซ้ำ โดยการใช้การถอดรหัสการเข้าถึงมาร์กกิงที่อยู่ในรูปแบบข้อความให้ออกมาอยู่ในรูปแบบของกราฟปริภูมิสถานะ ตัวอย่างการถอดรหัสการเข้าถึงมาร์กกิงอยู่ในรูปที่ 3.4 ตัวอย่างเช่น

- 1) บรรทัดที่ 2 (0, "S. 'p1'", 0) ภายในวงเล็บจะมีทั้งหมดสามลำดับ แสดงให้เห็นถึง
  - ลำดับแรกและลำดับที่สาม คือ 0 หมายถึง สถานะเริ่มต้น
  - ลำดับที่สอง คือ "S." ซึ่งแสดงให้เห็นว่าเป็นการอธิบายสถานะ ที่มีเพลส "p1" เกิดขึ้น
  - เมื่อรวมความหมายของทั้งหมด หมายความว่า สถานะเริ่มต้น คือ สถานะ s0 ที่มีเพลส p1
- 2) บรรทัดที่ 3 (0, "E. 't1'", 1) ภายในวงเล็บจะมีทั้งหมดสามลำดับ แสดงให้เห็นถึง
  - ลำดับแรก คือ 0 หมายถึง สถานะ s0
  - ลำดับที่สอง คือ "E." ซึ่งแสดงให้เห็นว่าเป็นการอธิบายทรานซิชัน ที่มีทรานซิชัน "t1"
  - ลำดับที่สาม คือ 1 หมายถึง สถานะ s1
  - เมื่อรวมความหมายของทั้งหมด หมายความว่า สถานะ s0 มีเส้นเชื่อมไปยังสถานะถัดไป คือ s1 โดยมีทรานซิชัน "t1" เป็นเงื่อนไขการเปลี่ยนแปลงสถานะ
- 3) บรรทัดที่ 4 (0, "E. 't2'", 2) ภายในวงเล็บจะมีทั้งหมดสามลำดับ แสดงให้เห็นถึง
  - ลำดับแรก คือ 0 หมายถึง สถานะ s0
  - ซึ่งแสดงให้เห็นว่าเป็นการอธิบายทรานซิชัน ที่มีทรานซิชัน "t2"
  - ลำดับที่สาม คือ 2 หมายถึง สถานะ s2

- เมื่อรวมความหมายของทั้งหมด หมายความว่า สถานะ  $s_0$  มีเส้นเชื่อมไปยังสถานะถัดไป คือ  $s_2$  โดยมีทรานซิชัน "t2" เป็นเงื่อนไขการเปลี่ยนแปลงสถานะ

อย่างไรก็ดี การเขียนเป็นกราฟปริภูมิสถานะไม่ได้สนใจทรานซิชันที่เป็นเงื่อนไขการเปลี่ยนแปลงสถานะ ดังนั้น กราฟปริภูมิสถานะจะไม่มีทรานซิชันปรากฏ จากตัวอย่างการถอดรหัสปริภูมิสถานะ ทำให้ทราบว่า จากสถานะ  $s_0$  สามารถไปสู่สถานะถัดไปได้สองสถานะ คือ สถานะ  $s_1$  และ  $s_2$  เมื่อถอดรหัสการเข้าถึงมาร์กกิงทั้งหมดจะได้เป็นกราฟปริภูมิสถานะที่อยู่ในรูปที่ 3.5



รูปที่ 3.5 ตัวอย่างการถอดรหัสการเข้าถึงมาร์กกิงทั้งหมดเพื่อให้ได้เป็นกราฟปริภูมิสถานะ

### 3.1.2 การสกัดเส้นทางย่อยจากกราฟปริภูมิสถานะ

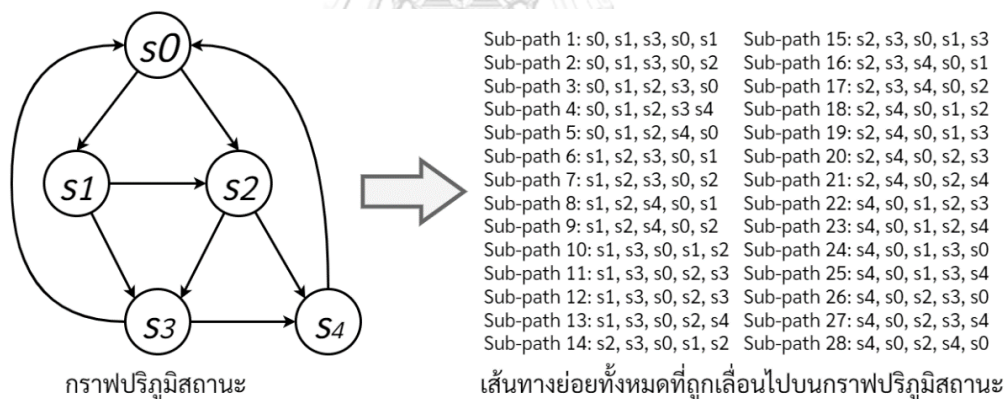
เมื่อได้กราฟปริภูมิสถานะจากขั้นตอนก่อนหน้าแล้ว ก็จะสามารถนำกราฟปริภูมิสถานะมาเลือกข้อมูลที่ถูกใช้ในการเรียนรู้ของแบบจำลองแอลเอสทีเอ็ม โดยมีรายละเอียดดังต่อไปนี้

เริ่มจากการสำรวจหาเส้นทางย่อย (Sub-path) บนกราฟปริภูมิสถานะ แต่เมื่อลองสังเกตบนกราฟปริภูมิสถานะแล้วจะพบว่ามี การวนซ้ำ ซึ่งเมื่อลองเขียนออกมาเป็นเส้นทางดำเนินงานจะมีบางเส้นทางดำเนินงานที่ไม่สิ้นสุด (Infinite execution paths) ทำให้ยากต่อการสร้างเส้นทางดำเนินงานที่เป็นไปได้ทั้งหมด

ดังนั้น เพื่อหลีกเลี่ยงปัญหานี้ ผู้วิจัยจึงใช้วิธีการหน้าต่างบานเลื่อน (Sliding window method) โดยการกำหนดขนาดของหน้าต่าง  $K$  สำหรับการเลื่อนบนเส้นทางดำเนินงาน เพื่อให้ได้เส้นทางย่อยที่เป็นไปได้ทั้งหมด ซึ่งค่า  $K$  หมายถึงจำนวนสถานะในเส้นทางย่อยและค่า  $K-1$  หมายถึงลำดับความยาวของเส้นทางย่อย ที่ใช้วิธีการเลื่อนไปบนกราฟตามจำนวนความยาวที่กำหนดที่ละสถานะโดยเริ่มจากสถานะเริ่มต้น เพื่อรวบรวมชุดของเส้นทางย่อยทั้งหมดเป็นข้อมูลสำหรับการเรียนรู้ของแบบจำลอง LSTM-MLC ซึ่งแต่ละเส้นทางย่อยจะมีความยาวเท่ากันและจะต้องมีลำดับไม่ซ้ำกัน

อย่างไรก็ตาม การกำหนดความยาว  $K$  นั้น ต้องถูกวิเคราะห์ เพื่อหาค่าความยาวที่เหมาะสมหรือมีภาระ (Overhead) น้อยที่สุดที่ทำให้แบบจำลองเรียนรู้ได้อย่างถูกต้อง เนื่องจาก หากระบุค่า  $K$  ที่มีความยาวมากๆ ทำให้เกิดเส้นทางย่อยที่ซ้ำซ้อนกันและแบบจำลองแอลเอสทีเอ็มใช้ความจำที่เกินความจำเป็น ซึ่งอาจนำไปสู่ปัญหาโอเวอร์ฟิตติ้งหรืออันเดอร์ฟิตติ้ง (Underfitting) [60] ในระหว่างกระบวนการเรียนรู้ ในทางกลับกัน หากกำหนดค่า  $K$  ที่น้อยเกินไปก็อาจส่งผลประสิทธิภาพการเรียนรู้ของแอลเอสทีเอ็ม

จากการทดลอง ผู้วิจัยได้ตั้งสมมติฐานที่ว่า ค่า  $K$  ควรเป็นค่าที่เล็กที่สุดที่ทำให้ได้ชุดข้อมูลที่สามารถนำไปให้แบบจำลอง LSTM-MLC เรียนรู้ได้อย่างมีประสิทธิภาพ โดยที่ผู้วิจัยได้มีการปรับเปลี่ยน  $K$  ต่างๆ และสังเกตประสิทธิภาพการเรียนรู้ของแบบจำลอง โดยผลการทดลอง พบว่า ค่า  $K$  ที่เหมาะสมในงานวิจัยนี้ คือ  $K = 5$  ซึ่งสามารถสกัดเส้นทางย่อยที่มีจำนวนน้อยที่สุดที่ทำให้แบบจำลอง LSTM-MLC เรียนรู้ได้อย่างมีประสิทธิภาพและสามารถนำไปใช้ได้กับทุกกรณีทดสอบของงานวิจัยนี้ และเมื่อนำมาสกัดเป็นเส้นทางย่อย หากเส้นทางย่อยใดมีความยาวน้อยกว่า  $K-1$  เส้นทางเหล่านั้นจะไม่ถูกนำมาใช้ ตัวอย่างเส้นทางย่อยที่เป็นไปได้ทั้งหมดที่ได้จากวิธีการหน้าต่างบานเลื่อนไปบนกราฟปริภูมิสถานะโดยกำหนดค่า  $K = 5$  แสดงในรูปที่ 3.6

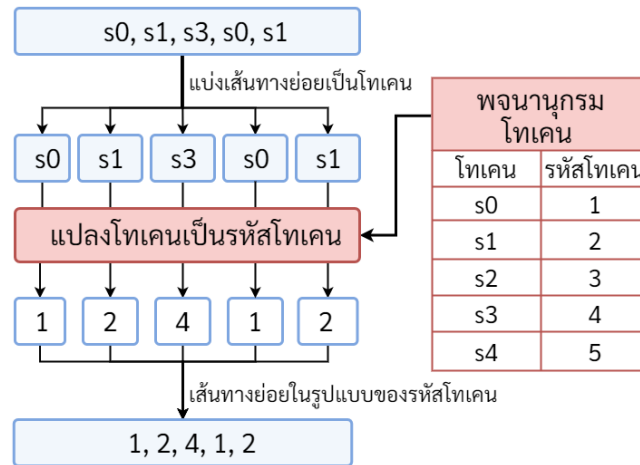


รูปที่ 3.6 ตัวอย่างเส้นทางย่อยที่เป็นไปได้ทั้งหมดที่ได้จากวิธีการหน้าต่างบานเลื่อน

### 3.1.3 การแปลงชื่อสถานะไปเป็นรหัสโทเคน

ในขั้นตอนนี้เป็นการแปลงข้อมูล ชุดข้อมูลจากขั้นตอนก่อนหน้าประกอบด้วยเส้นทางย่อยที่เป็นไปได้ทั้งหมดของกราฟปริภูมิสถานะถูกนำมาแปลงชื่อสถานะไปเป็นไปเป็น “รหัสโทเคน” (Token ID) โดยกระบวนการโทเคนไนเซชัน (Tokenization process) [61] เพื่อให้ได้ชุดข้อมูลที่อยู่ในรูปแบบที่ทำให้อัลกอริทึมของแมชชีนเลิร์นนิงสามารถเรียนรู้ได้ โดยที่หน่วยที่เล็กที่สุดเรียกว่า “โทเคน” ได้จากการแบ่งแยกสถานะออกมาจากเส้นทางย่อย

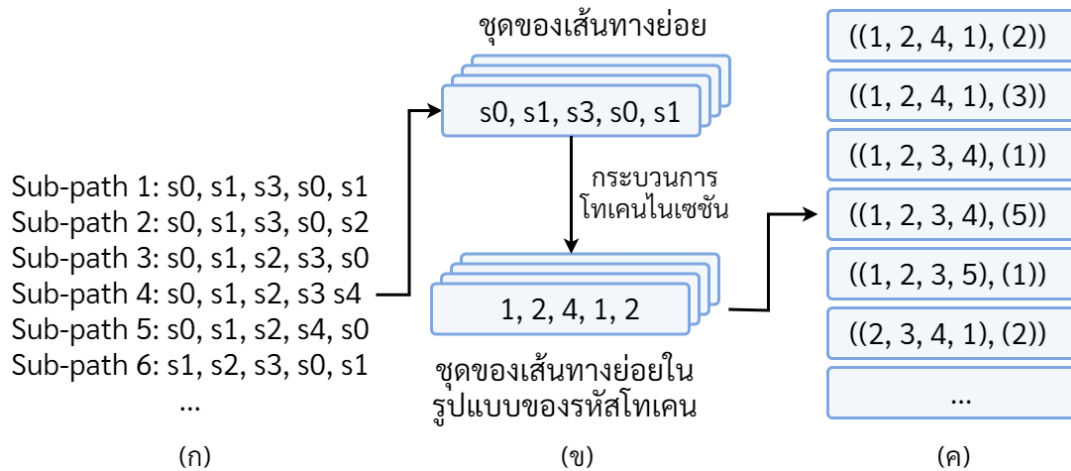
ในงานวิจัยนี้ โทเคนคือชื่อสถานะทั้งหมดที่อยู่บนกราฟปริภูมิสถานะและแต่ละรหัสโทเคนเป็นเลขจำนวนเต็มที่เป็นตัวแทนของชื่อสถานะที่ไม่ซ้ำกันที่ถูกรวบรวมในพจนานุกรมโทเคน (Tokenization dictionary) เมื่อผ่านกระบวนการโทเคนไนเซชัน เส้นทางย่อยที่แต่เดิมเป็นลำดับของชื่อสถานะถูกแปลงให้อยู่ในรูปแบบของลำดับของรหัสโทเคน กระบวนการทำโทเคนไนเซชันแสดงในรูปที่ 3.7 สำหรับ Sub-path 1: s0, s1, s3, s0, s1 ที่อยู่ในรูปที่ 3.6



รูปที่ 3.7 ตัวอย่างของกระบวนการทำโทเคนไนเซชันสำหรับ

### 3.1.4 การสร้างชุดข้อมูลสำหรับแบบจำลอง LSTM-MLC

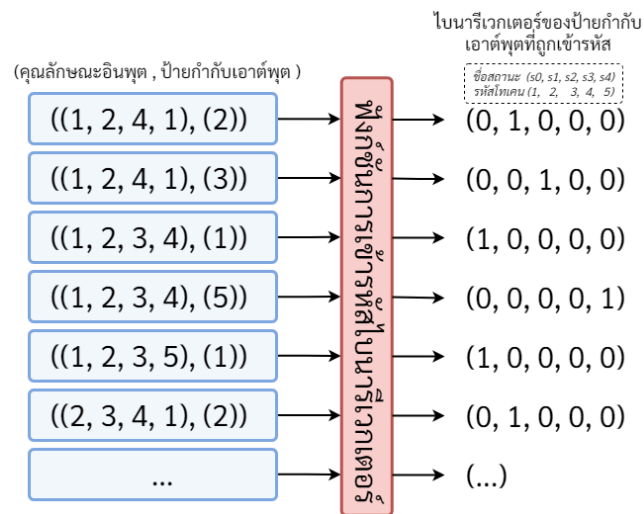
เมื่อได้ชุดข้อมูลประกอบด้วยเส้นทางย่อยที่อยู่ในรูปแบบของรหัสโทเคนเรียบร้อยแล้ว เส้นทางย่อยเหล่านั้นถูกนำไปสร้างเป็นชุดข้อมูลแบบการจำแนกแบบหลายป้าย โดยใช้การตัดลำดับท้ายสุดที่มีความยาวเท่ากับ  $K-1$  โดยที่  $K-1$  เป็นความยาวของเส้นทางย่อย ที่จะถูกกำหนดให้เป็นคุณลักษณะอินพุต (Input feature หรือ inf) และตัดตำแหน่งขวาสุดเป็นป้ายกำกับเอาต์พุต (Output label หรือ oul) ที่สอดคล้องกัน ซึ่งถูกเก็บให้เป็นชุดข้อมูลดิบ (Raw dataset) ที่อยู่ในรูปแบบคู่ลำดับของ (inf, oul)



รูปที่ 3.8 ตัวอย่างการตัดเส้นทางย่อยเพื่อให้ได้คุณลักษณะอินพุตและป้ายกำกับเอาต์พุต

ตัวอย่าง จากรูปที่ 3.7 “Sub-path 1: s0, s1, s3, s0, s1” ถูกแปลงให้เป็นรหัสโทเคน “1, 2, 4, 1, 2” โดยกระบวนการโทเคนไนเซชัน จากนั้น นำมาทำการแยกคุณลักษณะอินพุต คือ “1, 2, 4, 1” และป้ายกำกับเอาต์พุต คือ “2” ทำให้ได้ข้อมูลดิบจาก sub-path 1 คือ ((1, 2, 4, 1), (2)) และเมื่อทำไปจนครบทุกเส้นทางย่อยจะได้เป็นชุดข้อมูลดิบ แสดงวิธีการในรูปที่ 3.8 (ก) คือ เส้นทางย่อยทั้งหมด (ข) คือ กระบวนการนำชุดเส้นทางย่อยมาผ่านกระบวนการโทเคนไนเซชันและ (ค) คือ ชุดข้อมูลดิบที่ได้จากการแบ่งคุณลักษณะอินพุตและป้ายกำกับเอาต์พุต

อย่างไรก็ดี ป้ายกำกับเอาต์พุตที่อยู่ในชุดข้อมูลดิบถูกนำมาเข้ารหัสแบบวันฮอต (One-hot encoding) โดยการจัดการให้อยู่ในหมวดหมู่ เพื่อให้แบบจำลอง LSTM-MLC สามารถนำไปใช้ในการทำนายตามแต่ละหมวดหมู่ได้ง่าย ในงานวิจัยนี้ หมวดหมู่ คือ ชื่อของสถานะที่ถูกแปลงเป็นรหัสโทเคน ที่ถูกเข้ารหัสเป็นไบนารีเวกเตอร์ (Binary vector) ซึ่งจำนวนของหมวดหมู่เท่ากับจำนวนสถานะทั้งหมดที่แสดงอยู่บนกราฟปริภูมิสถานะ โดยที่ “1” หมายถึง ป้ายกำกับหมวดหมู่นั้นสอดคล้องกับคุณลักษณะอินพุต และ “0” หมายถึง ป้ายกำกับหมวดหมู่นั้นไม่สอดคล้องกับคุณลักษณะอินพุต ตัวอย่างจากรูปที่ 3.9 คุณลักษณะอินพุต (1, 2, 4, 1) มีป้ายกำกับเอาต์พุต (2) เมื่อนำไปผ่านฟังก์ชันการเข้ารหัสไบนารีเวกเตอร์แล้ว พบว่า “1” ปรากฏในตำแหน่งที่ 2 ของ (0, 1, 0, 0, 0) ซึ่งแสดงถึงป้ายกำกับหมวดหมู่ที่ 2 นั้นสอดคล้องกับคุณลักษณะ อินพุต (1, 2, 4, 1)



รูปที่ 3.9 ตัวอย่างการเข้ารหัสไบนารีเวกเตอร์สำหรับป้ายกำกับเอาต์พุต

จากคุณลักษณะของการจำแนกแบบหลายป้ายทำให้คุณลักษณะอินพุตสามารถมีได้มากกว่าหนึ่งป้ายกำกับเอาต์พุตหรือสามารถมีป้ายกำกับได้หลายหมวดหมู่ นั่น จำเป็นต้องมีการพิจารณาคุณลักษณะอินพุตที่เหมือนกัน โดยรวบรวมไบนารีเวกเตอร์เพื่อให้ได้ป้ายกำกับเอาต์พุตสำหรับการจำแนกแบบหลายป้าย ตัวอย่างเช่น ป้ายกำกับเอาต์พุตในรูปที่ 3.9 มีคุณลักษณะอินพุต  $(1, 2, 4, 1)$  เหมือนกันในบรรทัดที่ 1 และ 2 โดยมีป้ายกำกับเอาต์พุต (2) และ (3) และเมื่อนำไปผ่านการเข้ารหัสไบนารีเวกเตอร์ทำให้ได้  $(0, 1, 0, 0, 0)$  และ  $(0, 0, 1, 0, 0)$  ซึ่งจะถูกรวมเป็นไบนารีเวกเตอร์เดียวกันได้ คือ ไบนารีเวกเตอร์สำหรับป้ายกำกับเอาต์พุต  $(0, 1, 1, 0, 0)$  สำหรับคุณลักษณะอินพุต  $(1, 2, 4, 1)$  และเมื่อสังเกต พบว่ามีคุณลักษณะอินพุต  $(1, 2, 4, 1)$  เหมือนกันในบรรทัดที่ 3 และ 4 เช่นกัน ดังนั้นไบนารีเวกเตอร์สำหรับป้ายกำกับเอาต์พุต  $(1, 0, 0, 0, 0)$  และ  $(0, 0, 0, 0, 1)$  จึงถูกรวมเข้าด้วยกันเป็น  $(1, 0, 0, 0, 1)$  และสำหรับคุณลักษณะอินพุตที่ไม่มีตัวอื่นซ้ำก็ใช้ไบนารีเวกเตอร์สำหรับป้ายกำกับเอาต์พุตตัวนั้นได้เลย เมื่อพิจารณาคุณลักษณะอินพุตที่เหมือนกันทั้งหมด ก็ทำให้ได้เป็นชุดข้อมูลสำหรับแบบจำลอง LSTM-MLC จึงมีคุณลักษณะอินพุตและป้ายกำกับเอาต์พุตสำหรับการจำแนกแบบหลายป้าย ดังแสดงตัวอย่างได้ในรูปที่ 3.10

(คุณลักษณะอินพุต)	ป้ายกำกับเอาต์พุตสำหรับการจำแนกแบบหลายป้าย
(1, 2, 4, 1)	(0, 1, 1, 0, 0)
(1, 2, 3, 4)	(1, 0, 0, 0, 1)
(1, 2, 3, 5)	(1, 0, 0, 0, 0)
(2, 3, 4, 1)	(0, 1, 0, 0, 0)
...	...

รูปที่ 3.10 ตัวอย่างชุดข้อมูลสำหรับแบบจำลอง LSTM-MLC

จากการทดลองพบว่า หากข้อมูลสำหรับแบบจำลอง LSTM-MLC มีขนาดใหญ่ จะทำให้แบบจำลองใช้เวลาในการเรียนรู้มากและทำให้ไม่สามารถเห็นพฤติกรรมการเรียนรู้ของแบบจำลองได้อย่างทันทีทันใด ดังนั้น ในงานวิจัยนี้ได้นำชุดข้อมูลมาแบ่งออกเป็นก้อน (Chunk) โดยแต่ละก้อนมีการกำหนดขนาดความจุของข้อมูล โดยผู้วิจัยได้ทำการทดลอง เพื่อหาขนาดความจุของแต่ละก้อน ซึ่งในแต่ละรอบการเรียนรู้แบบจำลอง LSTM-MLC จำเป็นต้องทำการเรียนรู้ข้อมูลแต่ละก้อนให้ครบทุกก้อน เริ่มจากโหนดแบบจำลอง LSTM-MLC ขึ้นมา จากนั้นทำการเรียนรู้ข้อมูลก้อนนั้นและเมื่อเรียนรู้เสร็จสิ้นก็บันทึกแบบจำลอง จากนั้น ทำการเรียนรู้ข้อมูลก้อนถัดไปจนครบทุกก้อนโดยใช้วิธีการเดียวกัน ดังนั้น หากมีการกำหนดให้แต่ละก้อนมีขนาดความจุจำนวนน้อยก็ส่งผลให้เวลาเสียไปในการโหลดและบันทึกแบบจำลอง LSTM-MLC ด้วยเหตุนี้ ผู้วิจัยได้กำหนดให้ความจุของก้อนมีขนาดเท่ากับ 25,000 รายการ เป็นขนาดความจุที่ทำให้แบบจำลอง LSTM-MLC ใช้เวลาในการเรียนรู้ชุดข้อมูลของทุกกรณีศึกษาในงานวิจัยนี้ ซึ่งแต่ละรอบใช้เวลาไม่นานเท่าไรนักและทำให้สามารถสังเกตพฤติกรรมการเรียนรู้ของแบบจำลองได้ทันทีทันใด ตัวอย่างสมมติเช่น หากข้อมูลทั้งหมดมี 36,000 รายการ ข้อมูลนี้ถูกแบ่งออกเป็น 2 ก้อน คือ ข้อมูลรายการที่ 1 ถึง 25,000 ถูกนำมาใส่ในก้อนที่ 1 และข้อมูลรายการที่ 25,001 ถึง 36,000 ถูกนำมาใส่ในก้อนที่ 2 ดังนั้น ก็ทำให้ได้เป็นชุดข้อมูลที่ถูกแบ่งออกเป็น 2 ก้อน

หลังจากนั้น ชุดข้อมูลแต่ละก้อนถูกแบ่งเป็น 2 ชุดข้อมูล คือ ชุดข้อมูลการฝึกอบรม (Training dataset) ใช้สำหรับกระบวนการเรียนรู้ของแบบจำลอง LSTM-MLC และชุดข้อมูลการทดสอบ (Test dataset) ใช้สำหรับทดสอบความแม่นยำของแบบจำลอง LSTM-MLC ที่ได้รับการเรียนรู้เรียบร้อยแล้ว นอกจากนี้ในระหว่างการเรียนรู้ ชุดข้อมูลการเรียนรู้ถูกนำมาแบ่งข้อมูลออกเป็นชุดข้อมูลการตรวจสอบ (Validation dataset) โดยใช้วิธีการตรวจสอบไขว้ (K-Fold cross



validation) ซึ่งแต่ละส่วนมาจากการสุ่มเพื่อให้ข้อมูลกระจายเท่าๆ กัน สำหรับในแต่ละรอบการเรียนรู้เพื่อปรับให้แบบจำลองมีประสิทธิภาพการทำนายได้ดีขึ้น

### 3.2 การออกแบบและพัฒนาเครื่องสร้างปฏิภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอลทีแอลที่ใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้าย

ในขั้นตอนนี้ ผู้วิจัยได้ออกแบบวิธีการพัฒนาเครื่องสร้างปฏิภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอลทีแอลที่ใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้าย ซึ่งขอเรียกว่า เครื่องสร้างปฏิภูมิสถานะ โดยรายละเอียดดังต่อไปนี้

#### 3.2.1 การออกแบบและสร้างแบบจำลอง LSTM-MLC

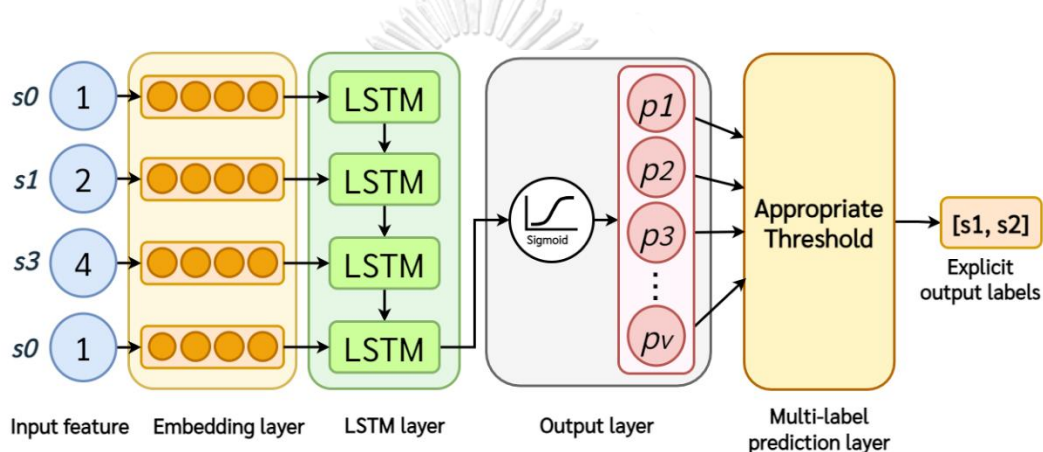
สำหรับการสร้างแบบจำลอง LSTM-MLC ผู้วิจัยได้ทำการค้นคว้าและทดลองว่าอัลกอริทึมของแมชชีนเลิร์นนิงประเภทใด มีความเหมาะสมกับลักษณะของข้อมูลในงานวิจัยนี้ ซึ่งจากการค้นคว้าจึงพบว่าแอลเอสทีเอ็มสามารถจดจำลำดับของข้อมูลและลักษณะของรูปแบบของการทำนายได้ดีหลายคำตอบ ดังนั้น งานวิจัยนี้เลือกใช้ประโยชน์จากแอลเอสทีเอ็มที่สามารถจำแนกแบบหลายป้ายเพื่อให้เหมาะสมกับข้อมูลสำหรับการทำนายสถานะถัดไปที่สอดคล้องกับเส้นทางดำเนินงานบนกราฟปฏิภูมิสถานะ

โครงสร้างสถาปัตยกรรมของของแบบจำลอง LSTM-MLC ในรูปที่ 3.11 มี 4 ชั้น (Layer) โดยที่ชั้นที่ 1-3 เป็นส่วนของโครงข่ายประสาทเทียมแอลเอสทีเอ็ม ในขณะที่ชั้นที่ 4 ผู้วิจัยได้เพิ่มเข้าไปเพื่อให้แบบจำลอง LSTM-MLC มีความสามารถในการจำแนกแบบหลายป้ายได้ ซึ่งรับข้อมูลนำเข้าคือ ลำดับของสถานะที่แสดงเป็นเส้นทางดำเนินงานบนกราฟปฏิภูมิสถานะและผลลัพธ์ที่ได้รับคือสถานะถัดไปที่สอดคล้องกับเส้นทางดำเนินงานที่รับเข้ามา โดยมีรายละเอียดดังต่อไปนี้

- 1) ชั้นการฝัง (Embedding layer) คือ ชั้นรับชุดข้อมูลสำหรับแบบจำลอง LSTM-MLC โดยมีมิติของชั้นนี้เท่ากับจำนวนของสถานะทั้งหมดของระบบที่สนใจและจำนวนข้อมูลนำเข้ามีความยาวเท่ากับขนาดของคุณลักษณะอินพุต
- 2) ชั้นแอลเอสทีเอ็ม (LSTM layer) คือ ชั้นแอลเอสทีเอ็มที่มีแอลเอสทีเอ็มเรียงต่อกันตามความยาวของข้อมูลนำเข้า แอลเอสทีเอ็มเริ่มประมวลผลข้อมูลตัวแรก ณ เวลา  $t_1$  และส่งผลการพิจารณาไปยังลำดับ  $t_2$ , จากนั้น พิจารณาและส่งผลลัพธ์ไปยังลำดับถัดไปเรื่อยๆ จนกระทั่งถึงลำดับสุดท้าย ณ เวลา  $t_4$  จากนั้นส่งผลลัพธ์การประมวลผลลำดับสุดท้ายไปยังชั้นถัดไป
- 3) ชั้นผลลัพธ์ (Output layer) คือ ชั้นที่รับผลจากฟังก์ชันซิกมอยด์ที่ผ่านการประมวลผลค่าผิดพลาด (Error) โดยใช้โบนารีครอสเอนโทรปี (Binary cross entropy) และมีการ

เลือกใช้ไฮเปอร์พารามิเตอร์ คือ เครื่องมือเพิ่มประสิทธิภาพอดัม (Adam optimizers) [62] และอัตราการเรียนรู้ (Learning rate) ซึ่งมีการปรับค่าตั้งแต่ 0.01 – 0.001 เพื่อให้แบบจำลองเรียนรู้ได้อย่างมีประสิทธิภาพ อย่างไรก็ตาม ผลลัพธ์ที่ได้จะยังคงอยู่ในรูปแบบของค่าความน่าจะเป็น ซึ่งยังไม่สามารถเลือกได้ป้ายหรือหมวดหมู่ใดเป็นผลลัพธ์การทำนายที่แท้จริง

- 4) ชั้นการจำแนกแบบหลายป้าย (Multi-label prediction layer) คือ เมื่อได้ผลลัพธ์ชั้นก่อนหน้า ผู้วิจัยได้เพิ่มชั้นการจำแนกแบบหลายป้าย ซึ่งผู้วิจัยได้ใช้วิธีการกำหนดขีดแบ่งที่เหมาะสมเพื่อใช้เลือกกว่าป้ายใดเป็นผลลัพธ์การทำนายที่แท้จริงของการทำนาย ซึ่งวิธีการกำหนดขีดแบ่งที่เหมาะสมจะกล่าวในหัวข้อถัดไป



รูปที่ 3.11 โครงสร้างสถาปัตยกรรมของแบบจำลอง LSTM-MLC

จากรูปที่ 3.11 กำหนดเส้นทางดำเนินงานที่ต้องการทำนายสถานะถัดไป คือ  $s_0, s_1, s_3$  และ  $s_0$  ตามลำดับ เมื่อทำการแปลงให้เป็นคุณลักษณะอินพุตได้ 1, 2, 4 และ 1 ซึ่งถูกส่งผ่านไปชั้นการฝังเพื่อทำการรวบรวมข้อมูล จากนั้น ส่งไปยังแอลเอสทีเอ็มชั้นผ่านตามกระบวนการของแอลเอสทีเอ็มซึ่งโหนดสุดท้ายเชื่อมไปยังชั้นผลลัพธ์เพื่อคำนวณนำผลลัพธ์จากแอลเอสทีเอ็มโดยใช้ฟังก์ชันซิกมอยด์โดยผลลัพธ์ที่อยู่ในชั้นผลลัพธ์ จากนั้น นำผลลัพธ์ผ่านการจำแนกแบบหลายป้ายเพื่อให้ได้ผลลัพธ์การทำนายที่แท้จริง ดังนั้นสรุปได้ว่า เมื่อป้อนข้อมูลนำเข้า โดยการกำหนดเป็นเส้นทางดำเนินงาน  $s_0, s_1, s_3$  และ  $s_0$  ผ่านแบบจำลอง LSTM-MLC เพื่อทำนายสถานะถัดไปจะได้ผลลัพธ์สถานะถัดไป คือสถานะ  $s_1$  และ  $s_2$

### 3.2.2 การเรียนรู้ของแบบจำลอง LSTM-MLC

กระบวนการเรียนรู้ของแบบจำลอง LSTM-MLC จะนำชุดข้อมูลการเรียนรู้ที่ได้จัดเตรียมไว้ก่อนหน้ามาใช้งาน ซึ่งในระหว่างการเรียนรู้ ผู้วิจัยได้พิจารณาผลการเรียนรู้ของแบบจำลอง หากพบว่า

แบบจำลองประสิทธิภาพในการเรียนรู้ไม่ดีนักหรือเกิดปัญหาโอเวอร์ฟิตติ้งหรืออันเดอร์ฟิตติ้งในระหว่างการเรียนรู้ จำเป็นต้องกลับไปยังขั้นตอนการจัดเตรียมข้อมูล เนื่องจาก จำนวนความยาวของข้อมูลนำเข้าแบบจำลอง LSTM-MLC ต้องมีขนาดเท่ากับ  $K-1$  ที่ เป็นความยาวของเส้นทางย่อยซึ่งเป็นข้อมูลนำเข้าที่ได้มาจากขั้นตอนก่อนหน้า ดังนั้น ผู้วิจัยได้ทำการทดลองหาค่า  $K$  ที่เหมาะสมที่ทำให้แบบจำลองสามารถเรียนรู้ได้อย่างถูกต้อง โดยผู้วิจัยได้ทำการทดลองปรับเปลี่ยนค่า  $K$  ในการสร้างชุดข้อมูลสำหรับการฝึกสอนแบบจำลอง LSTM-MLC และจากนั้นนำมาให้แบบจำลอง LSTM-MLC เรียนรู้ หากในระหว่างการเรียนแบบจำลองไม่สามารถเรียนรู้ได้ถูกต้อง ผู้วิจัยได้กลับไปทำการปรับค่า  $K$  ใหม่ จนกระทั่งได้เป็นค่า  $K = 5$  และได้  $K-1 = 4$  ที่ เป็นความยาวของแต่ละข้อมูลนำเข้าแบบจำลอง LSTM-MLC ที่เหมาะสมที่มีภาระน้อยที่สุดที่ทำให้แบบจำลองสามารถเรียนรู้ได้อย่างถูกต้อง เพราะหากมีการกำหนดค่า  $K$  ที่ไม่เหมาะสมก็อาจส่งผลกระทบต่อประสิทธิภาพการเรียนรู้ จากนั้นปรับแต่งจำนวนของข้อมูลที่ใช้เป็นชุดข้อมูลการฝึกอบรม เช่น หากชุดข้อมูลการเรียนรู้มีจำนวนน้อยเกินไปทำให้แบบจำลองไม่สามารถเรียนรู้ได้ จึงจำเป็นต้องเพิ่มจำนวนข้อมูลหรือปรับค่าไฮเปอร์พารามิเตอร์ของแบบจำลอง เช่น จำนวนโหนดของแอลเอสทีเอ็มหรืออัตราการเรียนรู้

เมื่อได้แบบจำลอง LSTM-MLC ที่มีประสิทธิภาพในการเรียนรู้แล้ว แบบจำลอง LSTM-MLC สามารถใช้หาขีดแบ่งที่เหมาะสม ที่ถูกกล่าวในหัวข้อถัดไป

### 3.2.3 การกำหนดขีดแบ่งที่เหมาะสมสำหรับแบบจำลอง LSTM-MLC

จากที่กล่าวก่อนหน้านี้ ตามโครงสร้างของแบบจำลอง LSTM-MLC จะส่งผลลัพธ์ที่อยู่ในรูปแบบของค่าความน่าจะเป็นของทุกป้าย ซึ่งยังไม่ใช่ผลการทำนายที่แท้จริง ดังนั้น ในงานวิจัยนี้จึงจำเป็นต้องหาวิธีการเลือกว่าป้ายใดเป็นผลการทำนายที่แท้จริง อย่างไรก็ตามวิธีการกำหนดขีดแบ่งมีหลายวิธี เช่น การเลือกใช้ค่าโดยปริยาย 0.5 การเลือกใช้ผลลัพธ์ที่มีค่ามากที่สุด น้อยที่สุดหรือค่าเฉลี่ยสำหรับทุกป้าย เป็นต้น ในงานวิจัยนี้ได้นำเสนอวิธีการกำหนดขีดแบ่งที่เหมาะสม ซึ่งเป็นค่าคงที่หนึ่งตัวเพื่อใช้เป็นขีดแบ่งสำหรับการจำแนกแบบหลายป้ายสำหรับแบบจำลอง LSTM-MLC ที่เรียนรู้ได้อย่างมีประสิทธิภาพ

ในกระบวนการกำหนดขีดแบ่งที่เหมาะสม เริ่มจากนำแบบจำลอง LSTM-MLC มาทำทำนายชุดข้อมูลการฝึกอบรมเพื่อหาเส้นโค้ง ROC จากนั้นใช้ทฤษฎีของ ยูเดน เจ-อินเด็กซ์ (Youden's J-Index) หรือเรียกว่า เจ-อินเด็กซ์ เพื่อหาเจ-อินเด็กซ์ที่มีค่ามากที่สุดเป็นจุดตัดของแต่ละป้ายที่เป็นตัวแทนของชื่อสถานะออกมา เมื่อได้จุดตัดของแต่ละป้ายออกมาแล้ว เลือกจุดตัดที่มีค่าน้อยที่สุดเป็นขีดแบ่งที่เหมาะสม หากมีจำนวนก้อนของข้อมูลมากกว่า 1 ก้อน จุดตัดที่น้อยที่สุดของแต่ละก้อนถูกนำมาคำนวณหาค่าเฉลี่ยเพื่อเลือกเป็นขีดแบ่งที่เหมาะสม

ตัวอย่างเช่น จุดตัดของแต่ละหมวดหมู่ที่ผ่านการหาพื้นที่ใต้กราฟเรียบร้อยแล้ว ซึ่งเป็นค่าสมมติที่ใช้ในตัวอย่างนี้เท่านั้น โดยหมวดหมู่ของสถานะ  $s_0$  ถึง  $s_4$  ได้แก่

- เจ-อินเด็กซ์ของสถานะ  $s_0 = 0.9655$
- เจ-อินเด็กซ์ของสถานะ  $s_1 = 0.9190$
- เจ-อินเด็กซ์ของสถานะ  $s_2 = 0.9999$
- เจ-อินเด็กซ์ของสถานะ  $s_3 = 0.9927$
- เจ-อินเด็กซ์ของสถานะ  $s_4 = 0.8898$

ซึ่งพบว่า จุดตัดของสถานะ  $s_4$  มีค่าที่น้อยที่สุดและข้อมูลทั้งหมด 1 ก้อน ดังนั้น จุดตัดของสถานะ  $s_4$  มีค่าที่น้อยที่สุด ถูกนำมาใช้เป็นขีดแบ่งที่เหมาะสมสำหรับการสร้างปริภูมิสถานะโดยใช้แบบจำลอง LSTM-MLC

อย่างไรก็ดี ขั้นตอนการหาขีดแบ่งนี้ทำโดยอัตโนมัติ หลังจากแบบจำลอง LSTM-MLC ผ่านการเรียนรู้แล้ว โดยไม่จำเป็นต้องกำหนดใหม่ทุกครั้ง แต่หากชุดข้อมูลมีการเปลี่ยนแปลงแบบจำลอง LSTM-MLC ต้องผ่านการฝึกอบรมใหม่ จำเป็นต้องทำการหาขีดแบ่งอีกครั้งเพื่อให้ได้แบบจำลองและขีดแบ่งที่เหมาะสมกับลักษณะชุดข้อมูล

### 3.2.4 การประเมินประสิทธิภาพการจำแนกหลายป้ายของแบบจำลอง LSTM-MLC

ในขั้นตอนนี้ แบบจำลอง LSTM-MLC ถูกนำมาทดสอบกับชุดข้อมูลการทดสอบที่ได้เตรียมไว้ โดยใช้ขีดแบ่งที่เหมาะสมเลือกผลลัพธ์การทำนายของแบบจำลองที่ถูกเปรียบเทียบกับค่าความจริงเพื่อวัดประสิทธิภาพของแบบจำลอง LSTM-MLC โดยใช้ตัววัดการประเมินผลของการจำแนกหลายป้ายที่ประกอบด้วย ค่าระยะที่หายไป (Hamming loss หรือ HL) ค่าความเที่ยงระดับไมโคร (Micro-Precision หรือ MP) ค่าเรียกคืนระดับไมโคร (Micro-Recall หรือ MR) และเอฟวันระดับไมโคร (Micro-F1 หรือ MF1) โดยค่า HL ที่คาดหวังต้องมีค่าน้อยมากหรือมีค่าเข้าใกล้ 0 ในขณะที่ ค่า MP MR และ MF1 ที่คาดหวังต้องมีค่าเข้าใกล้ 1 หรือ 100%

เมื่อพิจารณาการประเมินประสิทธิภาพการจำแนกแบบหลายป้ายของแบบจำลอง LSTM-MLC แล้วพบว่า ผลการประเมินไม่มีประสิทธิภาพตามที่คาดหวัง หมายความว่า แบบจำลอง LSTM-MLC และขีดแบ่งไม่มีประสิทธิภาพในการจำแนกชุดข้อมูลการทดสอบ ดังนั้น จำเป็นต้องกลับไปเริ่มตั้งแต่การจัดเตรียมข้อมูลใหม่อีกครั้ง แต่ในทางกลับกัน แบบจำลอง LSTM-MLC และขีดแบ่งมีประสิทธิภาพที่สามารถจำแนกแบบหลายป้ายได้ตามที่คาดหวัง คือ ค่า HL เข้าใกล้ 0 และมีค่า MP MR และ MF1 ที่มากกว่า 0.9 หรือ 90% ก็จะได้เป็นเครื่องสร้างปริภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอลทีแอลที่ใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้าย

ดังนั้น ส่วนประกอบหลักของเครื่องสร้างปฏิภูมิสถานะสำหรับโมเดลเช็กกิงแบบแอลทีแอลที่ใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้าย ประกอบด้วย แบบจำลอง LSTM-MLC และชุดแบ่งที่เหมาะสม

### 3.3 การสร้างปฏิภูมิสถานะโดยเครื่องสร้างปฏิภูมิสถานะ

เมื่อเครื่องสร้างปฏิภูมิสถานะสำหรับโมเดลเช็กกิงแบบแอลทีแอลที่ใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้ายถูกพัฒนาขึ้นมาแล้ว ทำให้สามารถนำเครื่องสร้างปฏิภูมิสถานะสร้างปฏิภูมิสถานะ โดยเริ่มจากการกำหนดเส้นทางดำเนินงานที่มีความยาวไม่เกิน  $K-1$  ซึ่งในงานวิจัยนี้ค่า  $K-1$  มีค่าเท่ากับ 4 ส่งเข้าไปให้เครื่องสร้างปฏิภูมิสถานะเริ่มทำนายหาสถานะถัดไปเพื่อรวบรวมเป็นเส้นทางดำเนินงานของปฏิภูมิสถานะของระบบที่เราสนใจ

ในรูปที่ 3.12 แสดงตัวอย่างการทำนายสถานะถัดไป โดยรับข้อมูลนำเข้า คือ เส้นทางดำเนินงานที่แสดงเป็นลำดับของสถานะ คือ  $s_0 s_1 s_2$  และ  $s_3$  ถูกส่งเข้าไปภายในเครื่องปฏิภูมิสถานะเพื่อทำนายสถานะถัดไป คือ สถานะ  $s_0$  และ  $s_4$  จากนั้น นำสถานะถัดไปมาเก็บเป็นลำดับเส้นทางดำเนินงานโดยการต่อท้ายลำดับสถานะที่เป็นข้อมูลนำเข้า จากตัวอย่างจะได้เส้นทางดำเนินงานเพิ่มเป็นสองเส้นทาง เนื่องจาก ผลการทำนายสถานะถัดไปที่แท้จริงมีทั้งหมดสองคำตอบ

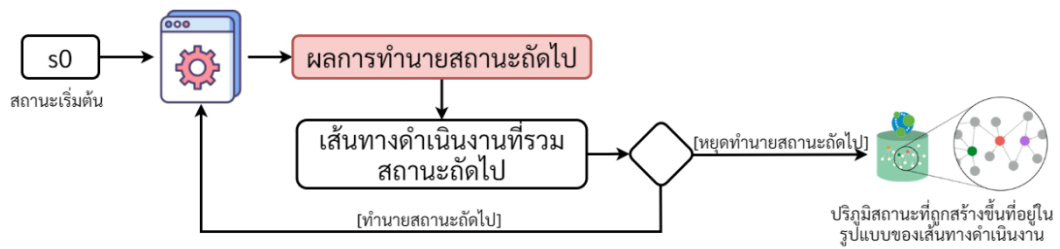


รูปที่ 3.12 ตัวอย่างการทำนายสถานะถัดไปโดยใช้เครื่องสร้างปฏิภูมิสถานะ

ดังนั้น ในกระบวนการนี้เป็นการนำเครื่องสร้างปฏิภูมิสถานะมาสร้างปฏิภูมิสถานะ โดยเริ่มทำนายจากสถานะเริ่มต้นไปที่ละสถานะและค่อยๆ เก็บเป็นลำดับของเส้นทางดำเนินงาน สุดท้ายเครื่องสร้างปฏิภูมิสถานะจะสามารถสร้างปฏิภูมิสถานะที่อยู่ในรูปแบบของชุดของเส้นทางดำเนินงานทั้งหมด

อย่างไรก็ดี การสร้างปฏิภูมิสถานะจะถูกกำหนดความยาวสำหรับการสร้างปฏิภูมิสถานะว่าจะหยุดสร้างเมื่อมีความยาวเท่าใด ซึ่งการกำหนดความยาวจะช่วยป้องกันการสร้างปฏิภูมิสถานะแบบไม่รู้จบ หากระบบมีการทำงานแบบวนซ้ำ ดังนั้น สำหรับการสร้างปฏิภูมิสถานะในงานวิจัยนี้ สามารถกำหนดความยาวได้ว่าจะหยุดเมื่อมีความยาวเท่าใดและสามารถกำหนดเส้นทางดำเนินงานที่ต้องการป้อนเข้าไปในเครื่องสร้างปฏิภูมิสถานะได้ตามที่ต้องการ โดยไม่จำเป็นต้องเริ่มที่สถานะเริ่มต้นเสมอ

แต่ขนาดของเส้นทางดำเนินงานต้องมีขนาดไม่เกิน  $K-1$  หรือความยาวไม่เกิน 4 จากนั้นแต่ละเส้นทางดำเนินงานถูกทำนายสถานะถัดไปจนกว่าความยาวของเส้นทางดำเนินงานเหล่านั้นมีความยาวตามที่กำหนด แต่หากมีบางเส้นทางดำเนินงานพบสถานะสุดท้าย (Final states) หรือพบการวนซ้ำแล้วเส้นทางดำเนินงานนั้นจะไม่ถูกนำมาทำนายสถานะถัดไป ซึ่งตัวอย่างการสร้างปริภูมิสถานะโดยเครื่องสร้างปริภูมิสถานะ แสดงในรูปที่ 3.13

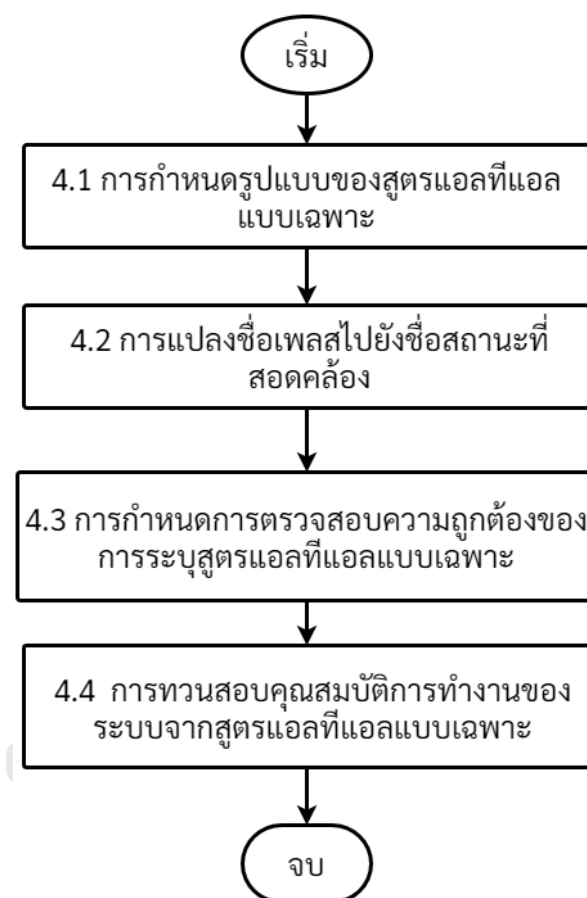


รูปที่ 3.13 ตัวอย่างการสร้างปริภูมิสถานะโดยเครื่องสร้างปริภูมิสถานะ

## บทที่ 4

### แนวคิดการพัฒนาเครื่องมือการทวนสอบคุณสมบัติโดยใช้เครื่องสร้างปริภูมิสถานะ

แนวคิดของงานวิจัยนี้ในบทนี้มีจุดประสงค์เพื่อสร้างเครื่องมือการทวนสอบคุณสมบัติการทำงานของระบบโดยนำความสามารถของเครื่องสร้างปริภูมิสถานะที่ได้จากบทที่ 3 มาใช้ในการทวนสอบคุณสมบัติการทำงานของระบบ ดังนั้น ผู้วิจัยได้วาดแผนภาพขั้นตอนการพัฒนาเครื่องมือการทวนสอบคุณสมบัติแบบเฉพาะ แสดงในรูปที่ 4.1 โดยมีรายละเอียดดังต่อไปนี้



รูปที่ 4.1 แผนภาพขั้นตอนการพัฒนาเครื่องมือการทวนสอบคุณสมบัติแบบเฉพาะ

จากรูปที่ 4.1 แสดงถึงเครื่องมือการทวนสอบที่ผู้วิจัยได้นำเสนอ เมื่อเปรียบเทียบกับกระบวนการทวนสอบด้วยโมเดลเช็กกิงแบบดั้งเดิมแสดงในรูปที่ 2.1 ทำให้เห็นความแตกต่างกัน คือ เครื่องมือการทวนสอบในงานวิจัยนี้สามารถรับสูตรแอลทีแอลจากผู้ใช้เพื่อมาทำการทวนสอบได้ทันที โดยที่จำเป็นต้องนำเข้าแบบจำลองเชิงรูปนัย ซึ่งต่างกับการทวนสอบโดยโมเดลเช็กกิงแบบดั้งเดิม เนื่องจาก ข้อมูลของแบบจำลองเชิงรูปนัยได้ถูกสอนไว้แบบจำลอง LSTM-MLC เรียบร้อยแล้ว จากนั้น ทำการทวนสอบเพื่อแสดงผลลัพธ์การทวนสอบของคุณสมบัติที่ถูกกำหนดโดยสูตรแอลทีแอล

ดังนั้น การทวนสอบคุณสมบัติการทำงานของระบบสำหรับการทำโมเดลเชิงกึ่งแบบแอลทีแอล เริ่มต้นรับข้อมูลนำเข้า คือ คุณสมบัติที่เขียนในรูปแบบของสูตรแอลทีแอล วิธีการทำงานของเครื่องมือการทวนสอบแบบทันทีหรือเรียกว่า “ออน-เดอะ-ฟลาย” (On-the-fly) ที่ใช้ความสามารถของเครื่องสร้างปริภูมิสถานะ เพื่อตรวจหารูปแบบของคุณสมบัติของแอลทีแอลภายในปริภูมิสถานะของระบบที่เราสนใจหรือไม่ ซึ่งผลการทวนสอบคุณสมบัติของระบบจะแสดงผลว่าเป็นที่พอใจ (Satisfied) หรือไม่พอใจ (Not satisfied) หากได้ผลไม่พอใจจะแสดงตัวอย่างค้าน (Counterexample)

#### 4.1 การกำหนดรูปแบบของสูตรแอลทีแอลแบบเฉพาะ

ขั้นตอนแรก กำหนดคุณสมบัติสำหรับการทวนสอบ ซึ่งเป็นคุณสมบัติพื้นฐานที่ระบบต้องทำการทวนสอบที่เขียนโดยใช้รูปแบบของสูตรแอลทีแอลแบบเฉพาะ (Specific LTL formula หรือ SLF) ประกอบด้วย ตัวดำเนินการเชิงเวลา G (Always) X (Next) F (Finally) และ U (Until) เท่านั้น สูตรแอลทีแอลแบบเฉพาะสำหรับคุณสมบัติที่ต้องการทวนสอบระบบประกอบด้วย คุณสมบัติด้านความปลอดภัย แทนด้วย “SLFsafety” ด้านความดำเนินชีวิต แทนด้วย “SLFlive” และด้านความสัมพันธ์ แทนด้วย “SLFcorre” นอกจากนี้ เพื่อให้ครอบคลุมตัวดำเนินการเชิงเวลาทั้งหมดมาใช้ให้เกิดประโยชน์ในงานวิจัยนี้ ผู้วิจัยได้เพิ่มรูปแบบสูตรแอลทีแอลประเภท Until “SLFuntil” ซึ่งสามารถอธิบายรายละเอียดได้ดังต่อไปนี้

- 1) รูปแบบสูตรแอลทีแอลของคุณสมบัติด้านความปลอดภัย SLFsafety ถูกเขียนแทนด้วย  $G\neg(\Phi_{\text{bad}})$  โดยที่ G คือตัวดำเนินการเชิงเวลา “เสมอ” (Always) และ  $\Phi_{\text{bad}}$  คือ สูตรแอลทีแอลที่แสดงถึงสถานการณ์ที่ไม่ดีที่จะไม่เกิดขึ้นในภายในระบบ
- 2) รูปแบบสูตรแอลทีแอลของคุณสมบัติด้านความดำเนินชีวิต SLFlive ถูกเขียนแทนด้วย  $F(\Phi_{\text{good}})$  โดยที่ F คือตัวดำเนินการเชิงเวลา “ในที่สุด” (Finally) และ  $\Phi_{\text{good}}$  คือ สูตรแอลทีแอลที่แสดงถึงสถานการณ์ที่ดีที่จะเกิดขึ้นในภายในระบบในที่สุด
- 3) รูปแบบสูตรแอลทีแอลของคุณสมบัติด้านความสัมพันธ์ SLFcorre ถูกเขียนแทนด้วย  $F(\Phi_{\text{hypo}}) \rightarrow F((\Phi_{\text{hypo}}) \wedge F(\Psi_{\text{conclus}}))$  โดยที่  $\Phi_{\text{hypo}}$  คือ สูตรแอลทีแอลของสถานการณ์ที่เป็นสมมติฐานและ  $\Psi_{\text{conclus}}$  คือ สูตรแอลทีแอลของสถานการณ์ที่เป็นข้อสรุป ซึ่งคุณสมบัติด้านความสัมพันธ์ หมายความว่า หากมีสถานการณ์สมมติฐานเกิดขึ้น ในที่สุดแล้วสถานการณ์ที่เป็นข้อสรุปก็จะเกิดขึ้นจริงในที่สุด
- 4) ในทำนองเดียวกัน SLFcorre สามารถถูกเขียนแทนด้วย  $F(\Phi_{\text{hypo}}) \rightarrow F((\Phi_{\text{hypo}}) \wedge X(\Psi_{\text{conclus}}))$  โดยที่ X คือตัวดำเนินการเชิงเวลา “ถัดไป” (Next) หมายความว่า หากมี



สถานการณ์สมมติฐานเกิดขึ้น ในที่สุดแล้วสถานการณ์ที่เป็นข้อสรุปก็จะเกิดขึ้นจริงในสถานะถัดไป

- 5) รูปแบบสูตรแอลที่แอลประเภท Until SLFuntil ถูกเขียนแทนด้วย  $\Phi \cup \Psi$  โดยที่  $\cup$  คือตัวดำเนินการเชิงเวลา “จนกระทั่ง” (Until)  $\Phi$  คือสูตรแอลที่แอลที่เป็นสถานการณ์แรก และ  $\Psi$  คือสูตรแอลที่แอลที่เป็นสถานการณ์ที่ต่อกลับ อย่างไรก็ตาม การใช้สูตรแอลที่แอลประเภท  $\cup$  มักถูกใช้ในการทวนสอบการเริ่มต้นของระบบ เมื่อมีสถานการณ์ ณ เวลาเริ่มต้นเป็นจริงแล้ว สถานการณ์  $\Phi$  จะดำเนินต่อไปเรื่อย ๆ จนกระทั่งมีสถานการณ์ที่คาดหวังมาต่อกลับ  $\Psi$  จากนั้น  $\Phi$  สามารถหยุดทำงานหรือยังคงทำงานต่อไปได้

สรุปได้ว่า งานวิจัยนี้ได้กำหนดกำหนดรูปแบบของสูตรแอลที่แอลแบบเฉพาะที่ครอบคลุมตัวเชื่อมประพจน์และตัวดำเนินการเชิงเวลา ได้ทั้งหมด 5 แบบ คือ

- 1) SLFsafety =  $G\neg(\Phi_{\text{bad}})$
- 2) SLFlive =  $F(\Phi_{\text{good}})$
- 3) SLFcorre =  $F(\Phi_{\text{hypo}}) \rightarrow F((\Phi_{\text{hypo}}) \wedge F(\Psi_{\text{conclus}}))$
- 4) SLFcorre =  $F(\Phi_{\text{hypo}}) \rightarrow F((\Phi_{\text{hypo}}) \wedge X(\Psi_{\text{conclus}}))$
- 5) SLFuntil =  $\Phi \cup \Psi$

#### 4.2 การแปลงชื่อเพลสไปยังชื่อสถานะที่สอดคล้อง

จากที่กล่าวก่อนหน้านี้ สูตรแอลที่แอลแบบเฉพาะที่ระบุประพจน์ด้วยชื่อของเพลสที่มีปรากฏในแบบจำลองเพทรีเน็ตเท่านั้น ดังนั้นก่อนการทวนสอบ เพลสที่อยู่ในสูตรแอลที่แอล จะถูกนำมาค้นหาว่าเพลสที่ต้องการทวนสอบปรากฏอยู่ในสถานะใดบ้าง โดยการแปลงชื่อเพลสไปยังชื่อสถานะที่เกี่ยวข้องทั้งหมด ด้วยเหตุนี้ ผู้วิจัยจึงได้กำหนดฟังก์ชันอีแอลเอฟ (ELF Function) คือ ฟังก์ชันการแมปเพลสไปยังสถานะ (Place-to-State mapping function) โดย  $ELF: P \rightarrow 2^S$  คือ แมปชื่อเพลสในแบบจำลองเพทรีเน็ตไปยังเซตของสถานะที่สอดคล้องกัน (Set of corresponding states หรือ CS)

ตัวอย่างเช่น เมื่อต้องการทวนสอบสูตรแอลที่แอลที่มีการกำหนดประพจน์ คือ red1 จากตัวอย่างระบบไฟจราจรสองต้นในรูปที่ 2.4 เริ่มค้นหาสถานะใดบ้างที่มีโทเคนบรรจุในเพลส red1 โดยใช้ฟังก์ชัน  $ELF(\text{red1})$  จะได้สถานะที่สอดคล้องกัน คือ สถานะ s0 s4 s7 s11 s15 และ s17 หรือเขียนใหม่ได้โดย  $CS = \{s0, s4, s7, s11, s15, s17\}$  โดยที่  $CS \subseteq S$  เมื่อนำทุกเพลสมาผ่านฟังก์ชันอีแอลเอฟ ก็จะได้ผลลัพธ์ดังรูปที่ 4.2

green1 : {s1, s3, s6, s9, s10, s14}  
 green2 : {s3, s5, s7, s10, s13, s15}  
 red1 : {s0, s4, s7, s11, s15, s17}  
 red2 : {s0, s1, s2, s4, s9, s12}  
 safe1 : {s0, s3, s5, s6, s7, s8, s9, s11, s12}  
 safe2 : {s1, s2, s4, s10, s13, s14, s15, s16, s17}  
 yellow1: {s2, s5, s8, s12, s13, s16}  
 yellow2: {s6, s8, s11, s14, s16, s17}

รูปที่ 4.2 ตัวอย่างผลลัพธ์ของฟังก์ชันอีแอลเอฟที่แมปชื่อเพลสไปยังสถานะที่สอดคล้องกัน

จากการกำหนดสูตรแอลทีแอลตามคุณสมบัติก่อนหน้า ประกอบด้วย  $\Phi_{\text{bad}}$   $\Phi_{\text{good}}$   $\Phi_{\text{hypo}}$  และ  $\Psi_{\text{conclus}}$  สามารถระบุประพจน์ที่เป็นชื่อเพลสได้มากกว่า 1 เพลส โดยเชื่อมต่อกันโดยใช้ “และ” (Conjunction) แทนด้วย  $\wedge$  และ “หรือ” (Disjunction) แทนด้วย  $\vee$  อีกทั้งยังสามารถระบุ นิเสธ (Negation) แทนด้วย  $\neg$  ที่หน้าชื่อเพลสได้ จากรูปที่ 4.2 สามารถอธิบายรายละเอียดได้ดังต่อไปนี้

- 1) หากมีเพลสเชื่อมต่อกันด้วยด้วยเครื่องหมาย  $\wedge$  หมายความว่า ต้องพิจารณาจากสถานะที่มีร่วมกัน โดยใช้การหาส่วนร่วม (Intersection) ตัวอย่างเช่น ถ้า  $F(\text{green1} \wedge \text{green2})$  แล้ว  $CS_1 = \text{ELF}(\text{green1}) \cap \text{ELF}(\text{green2})$  จะได้  $CS_1 = \{s3, s10\}$
- 2) หากมีเพลสเชื่อมต่อกันด้วยด้วยเครื่องหมาย  $\vee$  หมายความว่า ต้องพิจารณาจากสถานะทั้งหมดของทุกเพลส โดยใช้การหาผลรวม (Union) ตัวอย่างเช่น ถ้า  $F(\text{red1} \vee \text{red2})$  แล้ว  $CS_2 = \text{ELF}(\text{red1}) \cup \text{ELF}(\text{red2})$  จะได้  $CS_2 = \{s0, s1, s2, s4, s7, s9, s11, s12, s15, s17\}$
- 3) หากเพลสมีตัวนำหน้าด้วยเครื่องหมาย  $\neg$  หมายความว่า ต้องไม่พิจารณาสถานะที่สอดคล้องกันของเพลสนั้น ตัวอย่างเช่น ถ้า  $F(\neg\text{green2})$  แล้ว  $CS_3 = \text{ELF}(\text{green2})$  ดังนั้น  $CS_3 = \xi - \{s1, s3, s6, s9, s10, s14\}$  โดยที่  $\xi$  คือ เอกภพสัมพัทธ์ (Universal set) ของชุดสถานะทั้งหมดของระบบ

ในทางปฏิบัติ สถานะที่อยู่ในชุดของสถานะที่สอดคล้องกัน  $CS$  ถูกพิจารณาโดยการเลือกด้วยสัญลักษณ์  $\vee$  ตัวอย่างเช่น  $F(\text{green1} \wedge \text{green2})$  จะได้  $CS_1 = \{s3, s10\}$  จากนั้น  $CS_1$  ถูกพิจารณาเป็น  $s3 \vee s10$  ซึ่งสามารถเขียนใหม่ ให้อยู่ในรูปแบบของสูตรแอลทีแอลที่แปลงประพจน์ที่เป็นชื่อเพลสไปยังชื่อสถานะได้คือ  $F(s3 \vee s10)$  ทำให้ได้สูตรแอลทีแอลไปใช้ในการทวนสอบโดยเครื่องมือการทวนสอบในงานวิจัยนี้ ตัวอย่างเพิ่มเติมสามารถแสดงได้ในตารางที่ 4.1

ตารางที่ 4.1 ตัวอย่างการกำหนดสูตรแอลทีแอลทีที่ใช้ในการทวนสอบ

สูตรแอลทีแอลทีที่กำหนด โดยระบุประพจน์เป็นชื่อ เฟลส	สูตรแอลทีแอลทีที่ได้จากการแปลงประพจน์ที่เป็นชื่อเฟลสไปยังชื่อ สถานะ
$G \neg(\text{green1} \wedge \text{green2})$	$G \neg(s3 \vee s10)$
$G \neg(\text{safe1} \wedge \text{safe2})$	$G \neg(\emptyset)$
$F(\text{green1} \wedge \text{green2})$	$F(s3 \vee s10)$
$F(\text{red1} \vee \text{red2})$	$F(s0 \vee s1 \vee s2 \vee s4 \vee s7 \vee s9 \vee s11 \vee s12 \vee s15 \vee s17)$
$F \neg(\text{green2})$	$F(\neg s1 \vee \neg s3 \vee \neg s6 \vee \neg s9 \vee \neg s10 \vee \neg s14)$
$F(\text{red1}) \rightarrow F((\text{red1}) \wedge F(\text{green1}))$	$F(s0 \vee s4 \vee s7 \vee s11 \vee s15 \vee s17) \rightarrow F((s0 \vee s4 \vee s7 \vee s11 \vee s15 \vee s17) \vee F(s1 \vee s3 \vee s6 \vee s9 \vee s10 \vee s14))$
$\text{red1} \cup \text{green1}$	$(s0 \vee s4 \vee s7 \vee s1 \vee s15 \vee s17) \cup (s1 \vee s3 \vee s6 \vee s9 \vee s10 \vee s14)$

#### 4.3 การกำหนดการตรวจสอบความถูกต้องของการระบุสูตรแอลทีแอลแบบเฉพาะ

เมื่อได้กำหนดรูปแบบของสูตรแอลทีแอลเรียบร้อยแล้ว ผู้วิจัยได้ทำการตรวจสอบข้อมูลนำเข้า (Input Validation) ก่อนนำสูตรแอลทีแอลไปทำการทวนสอบโดยเครื่องมือการทวนสอบคุณสมบัติการทำงานของระบบ หากมีการกำหนดสูตรแอลทีแอลที่ไม่เป็นตามข้อกำหนด สูตรแอลทีแอลนั้นจะไม่ถูกนำไปใช้ในการทวนสอบ โดยรายละเอียดดังต่อไปนี้

- 1) ตรวจสอบความสมมูลของเครื่องหมายวงเล็บ
- 2) ตรวจสอบชื่อเฟลสว่ามีอยู่ในแบบจำลองเพทรีเน็ตที่จำลองระบบที่สนใจหรือไม่
- 3) ตรวจสอบวิธีการระบุโทเคน หากต้องการระบุจำนวนโทเคนมีมากกว่า 1 ต้องใช้เครื่องหมาย “\*” และตามด้วยตัวเลข เช่น  $p3^*3$  หมายความว่า เฟลส  $p3$  มีโทเคน 3 อัน
- 4) ตรวจสอบตัวเชื่อมประพจน์ สามารถเชื่อมด้วย “และ” “หรือ” เท่านั้น และสามารถใส่สัญลักษณ์อื่นๆ ได้ดังนี้
  - 4.1) “และ” (Conjunction) ใช้เครื่องหมาย “&&” แทนการเขียนด้วย  $\wedge$
  - 4.2) “หรือ” (Disjunction) ใช้เครื่องหมาย “||” แทนการเขียนด้วย  $\vee$
  - 4.3) นิเสธ (Negation) ใช้เครื่องหมาย “-” แทนการเขียนด้วย  $\neg$
  - 4.4) การให้เหตุผล (Implication) ใช้เครื่องหมาย “=>” แทนการเขียนด้วย  $\rightarrow$

- 4.5) ตัวเชื่อมประพจน์สามารถใช้ตัวใดตัวหนึ่งเท่านั้น เช่น  $(p1 \ \&\& \ p2 \ \&\& \ p3)$  หรือ  $(p1 \ \parallel \ p2 \ \parallel \ p3)$  ไม่สามารถใช้ตัวเชื่อมประพจน์รวมกันได้ เช่น  $(p1 \ \&\& \ p2 \ \parallel \ p3)$
- 5) ตรวจสอบตัวดำเนินการเชิงเวลาที่รองรับ ได้แก่
- 5.1) เสมอ (Always) แทนด้วยตัวอักษร G
  - 5.2) ถัดไป (Next) แทนด้วยตัวอักษร X
  - 5.3) ในที่สุด (Finally) แทนด้วยตัวอักษร F
  - 5.4) จนกระทั่ง (Until) แทนด้วยตัวอักษร U
- 6) สูตรแอลที่แอลสำหรับแต่ละคุณสมบัติที่ต้องการทวนสอบต้องอยู่ในรูปแบบเฉพาะที่กำหนดไว้เท่านั้น คือ
- 6.1) ด้านความปลอดภัย  $SLFsafety = G(\neg \Phi_{bad})$
  - 6.2) ด้านความดำเนินชีวิต  $SLFlive = F(\Phi_{good})$
  - 6.3) ด้านความสัมพันธ์  $SLFcorre = F(\Phi_{hypo}) \Rightarrow F((\Phi_{hypo}) \ \&\& \ F(\Psi_{conclus}))$
  - 6.4) ด้านความสัมพันธ์  $SLFcorre = F(\Phi_{hypo}) \Rightarrow F((\Phi_{hypo}) \ \&\& \ X(\Psi_{conclus}))$
  - 6.5) สูตรแอลที่แอลประเภท  $SLFuntil = \Phi \ U \ \Psi$

ในตารางที่ 4.2 แสดงตัวอย่างของผลลัพธ์การตรวจสอบข้อมูลนำเข้าที่เป็นสูตรแอลที่แอลแบบเฉพาะสำหรับเครื่องมือการทวนสอบคุณสมบัติของระบบ โดยชื่อของเพลสที่ระบุเป็นการสมมติขึ้นมาเท่านั้น โดยที่ตัวหนังสือสีแดงแสดงให้เห็นได้ชัดเจนถึงรูปแบบที่ไม่ถูกต้อง

ตารางที่ 4.2 ตัวอย่างของผลลัพธ์การตรวจสอบข้อมูลนำเข้าที่เป็นสูตรแอลที่แอลแบบเฉพาะ

คุณสมบัติ	สูตรแอลที่แอลแบบเฉพาะ	ผลลัพธ์การตรวจสอบ
ความปลอดภัย	$G(p0)$	เครื่องหมายวงเล็บไม่สมดุล
ความปลอดภัย	$A(p0 \ \parallel \ p1)$	พบตัวดำเนินการเชิงเวลาที่ไม่รองรับใน $A(p0 \ \parallel \ p1)$
ความปลอดภัย	$G(p0 \ \wedge \ p1)$	ไม่รองรับตัวเชื่อมประพจน์ ใน $G(p0 \ \wedge \ p1)$
ความดำเนินชีวิต	$F(p0 \ \&\& \ p1 \ \parallel \ p2)$	ไม่รับรองการระบุตัวเชื่อมประพจน์หลายตัว
ความดำเนินชีวิต	$F(abc)$	ไม่รู้จักตัวแปรประพจน์ “abc”
ความสัมพันธ์	$F(p0) \Rightarrow F(p1) \ \&\& \ F(p2)$	ชื่อเพลสสมมติฐานไม่สอดคล้องกัน
ความสัมพันธ์	$F(p0) \Rightarrow F(p0) \ \&\& \ G(p2)$	พบตัวดำเนินการเชิงเวลาที่ไม่รองรับใน $F(p0) \Rightarrow F(p1) \ \&\& \ G(p2)$

#### 4.4 การทวนสอบคุณสมบัติการทำงานของระบบจากสูตรแอลทีแอลแบบเฉพาะ

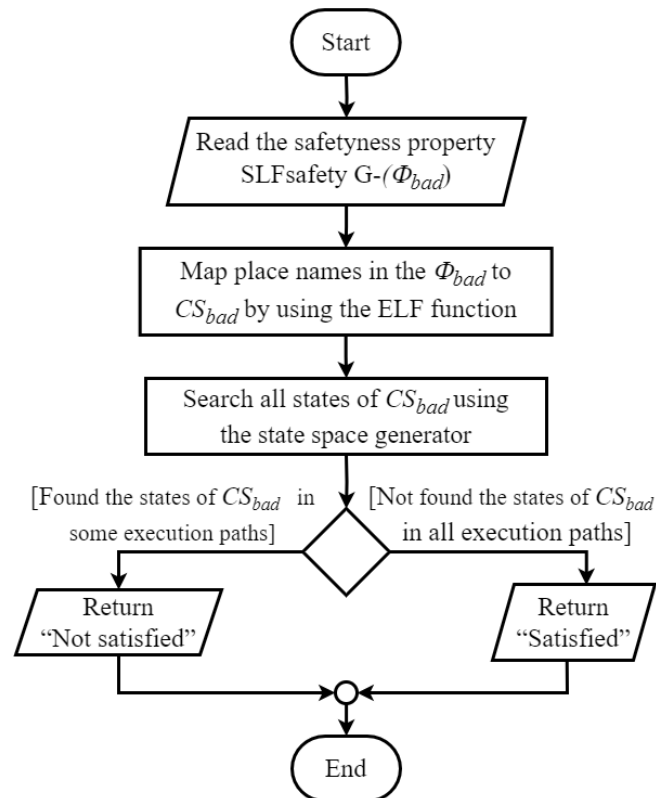
เมื่อได้สูตรแอลทีแอลที่ถูกแปลงชื่อเพลสไปยังชื่อสถานะที่สอดคล้องกันแล้ว จากนั้นสูตรแอลทีแอลถูกนำมาทวนสอบคุณสมบัติการทำงานของระบบ โดยใช้เครื่องมือการทวนสอบที่มีการเรียกใช้เครื่องสร้างปริภูมิสถานะในการสร้างเส้นทางดำเนินงาน เพื่อค้นหาสถานะที่คาดหวัง โดยเทคนิคแบบออน-เดอะ-ฟลาย โดยเริ่มค้นหาตั้งแต่สถานะเริ่มต้น  $s_0$  เสมอ ไปจนถึงจำนวนความลึกของเส้นทางดำเนินงานที่ต้องการทวนสอบ หากกำหนดความลึกที่มีจำนวนมากก็อาจทำให้ใช้เวลาในการทวนสอบนาน โดยมีขั้นตอนการทวนสอบดังต่อไปนี้

- 1) เริ่มจากอ่านสูตรแอลทีแอลที่ผู้ใช้กำหนดเข้ามาตามคุณสมบัติที่ต้องการทวนสอบ
- 2) ใช้ฟังก์ชันอีแอลเอฟแปลงชื่อเพลสไปยังชื่อสถานะที่สอดคล้องกันและตีความสูตรแอลทีแอลเพื่อให้ได้เป็นสูตรแอลทีแอลที่ถูกแปลงชื่อเพลสไปยังชื่อสถานะที่สอดคล้องกัน
- 3) นำเครื่องสร้างปริภูมิสถานะมาสร้างเส้นทางดำเนินงาน โดยเริ่มจาก สถานะเริ่มต้น  $s_0$  ที่ละหนึ่งรอบ จากนั้นจะได้เส้นทางดำเนินงานที่ถูกสร้างออกมา
- 4) นำเส้นทางดำเนินงานที่ถูกสร้างออกมา ทวนสอบตามคุณสมบัติที่ผู้ใช้กำหนดเข้ามา หากพบสถานะตามเงื่อนไขการทวนสอบของคุณสมบัติ ก็สามารถส่งผลการทวนสอบได้ทันที
- 5) หากยังไม่พบคุณสมบัติที่ต้องการ กลับไปยังข้อ 3) เพื่อให้เครื่องสร้างปริภูมิสถานะสร้างเส้นทางอีกครั้งและ ในข้อ 4) นำเส้นทางเหล่านั้นมาทำการทวนสอบอีกครั้ง และทำแบบนี้ไปเรื่อย ๆ จนถึงความลึกที่ผู้ใช้ต้องการทวนสอบ ซึ่งความลึกนี้ผู้ใช้สามารถกำหนดผ่านระบบที่งานวิจัยนี้พัฒนาขึ้นมาได้

อย่างไรก็ดี วิธีการทวนสอบแต่ละคุณสมบัติมีเงื่อนไขการทวนสอบที่แตกต่างกัน ดังนั้น ผู้วิจัยได้อธิบายเงื่อนไขของการทวนสอบของแต่ละคุณสมบัติได้ดังต่อไปนี้

##### 4.4.1 การทวนสอบคุณสมบัติด้านความปลอดภัย

กระบวนการทวนสอบคุณสมบัติด้านความปลอดภัย เริ่มจากการอ่านสูตรแอลทีแอลแบบเฉพาะ คือ  $SLF_{safety} = G-(\Phi_{bad})$  จากนั้น ใช้ฟังก์ชันอีแอลเอฟแปลงชื่อเพลสไปยังชื่อสถานะที่สอดคล้องกัน คือ  $CS_{bad}$  ซึ่งถือว่าเป็นสถานะที่ไม่ดีหรือเป็นสถานะที่คาดหวังว่าจะไม่เกิดขึ้นภายในระบบ สามารถแสดงเป็นแผนภาพกิจกรรมได้ในรูปที่ 4.3

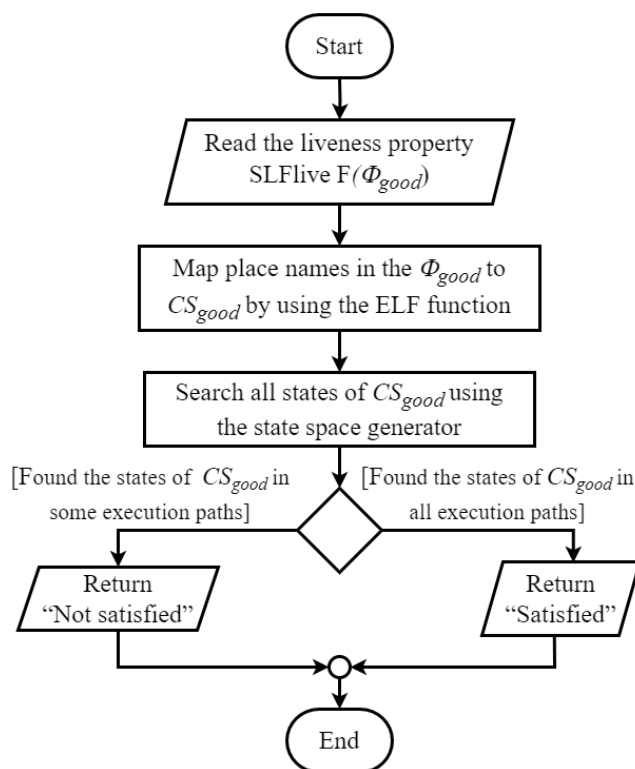


รูปที่ 4.3 แผนภาพกิจกรรมการทวนสอบคุณสมบัติด้านความปลอดภัย

จากรูปที่ 4.3 เครื่องมือการทวนสอบเริ่มค้นหาทุกสถานะของ  $CS_{bad}$  จากเส้นทางดำเนินงานที่ถูกสร้างออกมาโดยเครื่องสร้างปริภูมิสถานะ โดยเริ่มจากสถานะเริ่มต้น  $s_0$  หากไม่พบทุกสถานะของ  $CS_{bad}$  บนทุกเส้นทางดำเนินงานได้ผลลัพธ์เป็นที่พอใจ ในทางกลับกัน หากพบบางสถานะที่ไม่ติดบนบางเส้นทางดำเนินงาน ผลลัพธ์จะเป็นที่ไม่พอใจ

#### 4.4.2 การทวนสอบคุณสมบัติด้านความดำเนินชีวิต

กระบวนการทวนสอบคุณสมบัติด้านความดำเนินชีวิต เริ่มจากการอ่านสูตรแอลที่แอลแบบเฉพาะ คือ  $SLF_{live} = F(\Phi_{good})$  จากนั้นแปลงชื่อเพลสไปยังชื่อสถานะที่สอดคล้องกัน คือ  $CS_{good}$  ซึ่งถือว่าเป็นสถานะที่ดีหรือเป็นสถานะที่คาดหวังว่าจะเกิดขึ้นภายในระบบ ซึ่งทุกสถานะของ  $CS_{good}$  จะต้องได้รับการตรวจสอบ สามารถแสดงเป็นแผนภาพกิจกรรมได้ในรูปที่ 4.4

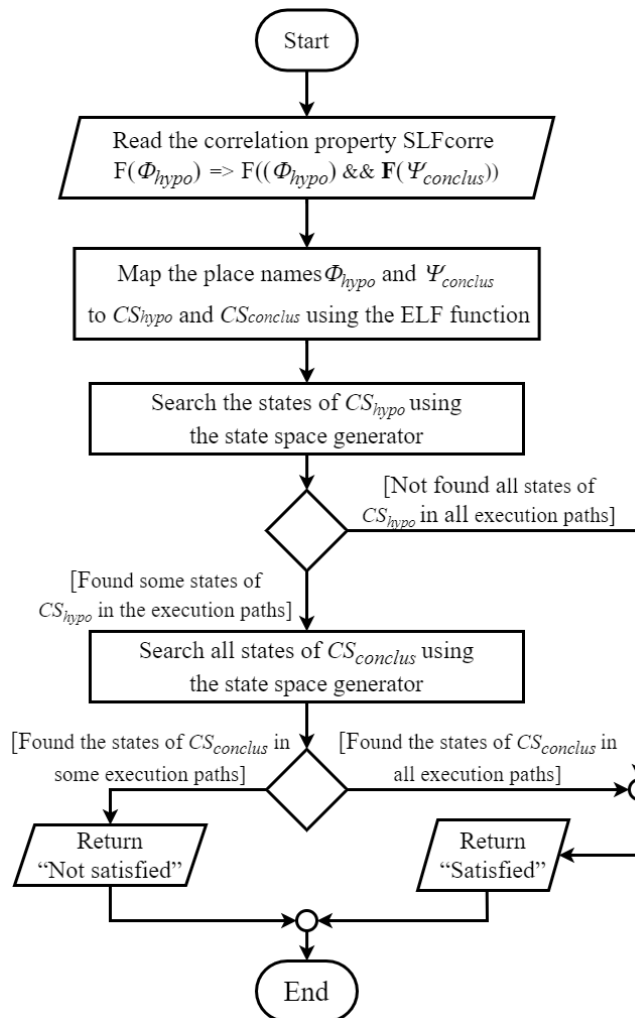


รูปที่ 4.4 แผนภาพกิจกรรมการทวนสอบคุณสมบัติด้านความดำเนินชีวิต

จากรูปที่ 4.4 เครื่องมือการทวนสอบเริ่มค้นหาทุกสถานะของ  $CS_{good}$  จากทุกเส้นทางดำเนินงานที่สร้างออกมาโดยเครื่องสร้างปริภูมิสถานะ โดยเริ่มจากสถานะเริ่มต้น  $s_0$  หากทุกเส้นทางดำเนินงานมีสถานะที่ดี  $CS_{good}$  เกิดขึ้นทั้งหมดจะได้ผลลัพธ์เป็นที่พอใจ ในทางกลับกัน หากมีบางเส้นทางดำเนินงานไม่พบสถานะที่ดี ผลลัพธ์จะเป็นที่ไม่พอใจ

#### 4.4.3 การทวนสอบคุณสมบัติด้านความสัมพันธ์

กระบวนการทวนสอบคุณสมบัติด้านความสัมพันธ์ เริ่มจากการอ่านสูตรแอลทีแอลแบบเฉพาะ คือ  $SLF_{corre} = F(\Phi_{hypo}) \Rightarrow F((\Phi_{hypo}) \ \&\& \ F(\Psi_{conclus}))$  จากนั้น แปลงชื่อเพลสไปยังชื่อสถานะที่สอดคล้องกัน คือ  $CS_{hypo}$  เป็นสถานะสมมติฐาน และ  $CS_{conclus}$  เป็นสถานะข้อสรุป สามารถแสดงเป็นแผนภาพกิจกรรม ดังรูปที่ 4.5

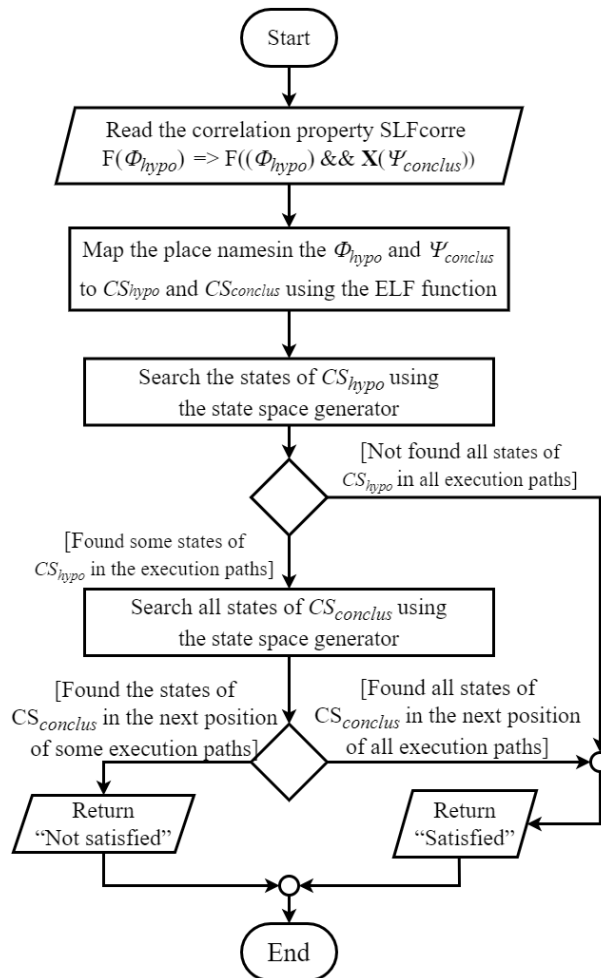


รูปที่ 4.5 แผนภาพกิจกรรมการทวนสอบคุณสมบัติด้านความสัมพันธ์แบบที่หนึ่ง

จากรูปที่ 4.5 เครื่องมือการทวนสอบเริ่มค้นหาทุกสถานะของ  $CS_{hypo}$  จากทุกเส้นทางดำเนินงานที่สร้างออกมาโดยเครื่องสร้างปริภูมิสถานะ โดยเริ่มจากสถานะเริ่มต้น  $s_0$  หากพบสถานะสมมติฐานแล้ว จากนั้นเริ่มทำการค้นหาสถานะข้อสรุปบนเส้นทางดำเนินงานที่พบสถานะสมมติฐานเมื่อพบสถานะข้อสรุปบนทุกเส้นทางดำเนินงานแล้วจะได้ผลลัพธ์เป็นที่พอใจ ในทำนองเดียวกัน หากไม่พบสถานะสมมติฐานในทุกเส้นทางดำเนินงานก็จะได้ผลลัพธ์เป็นที่พอใจเช่นกัน ในทางกลับกัน หากพบสถานะสมมติฐานแล้ว แต่พบสถานะข้อสรุปในบางเส้นทางดำเนินงานสำหรับเส้นทางที่พบสถานะสมมติฐานแล้วผลลัพธ์จะเป็นที่ไม่พอใจ

นอกจากนี้ SLFcorre สามารถเขียนได้อีกแบบ คือ  $SLFcorre = F(\Phi_{hypo}) => F((\Phi_{hypo}) \&\& X(\Psi_{conclus}))$  ซึ่งในกรณีนี้จะตรวจสอบ โดยแปลงชื่อเพลสไปยังชื่อสถานะที่สอดคล้องกัน เช่นเดียวกัน สามารถแสดงเป็นแผนภาพกิจกรรมดังรูปที่ 4.6



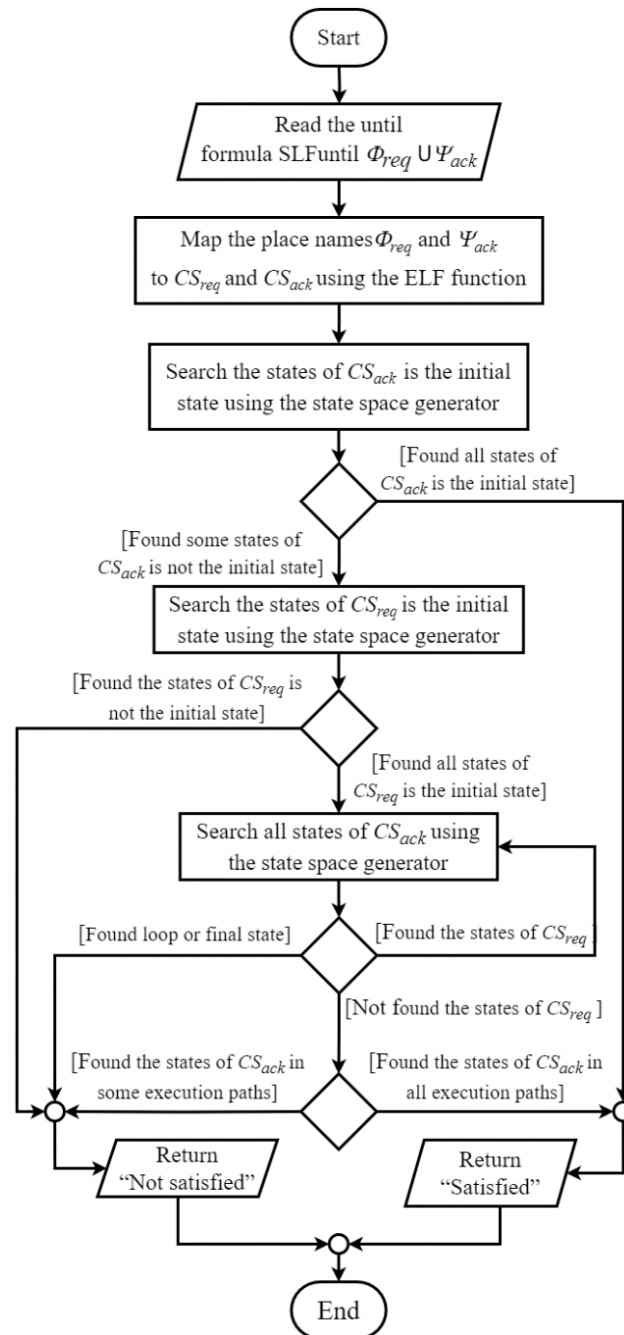


รูปที่ 4.6 แผนภาพกิจกรรมการทดสอบคุณสมบัติด้านความสัมพันธ์แบบที่สอง

จากรูปที่ 4.6 เครื่องมือการทดสอบ เริ่มค้นหาสถานะของ  $CS_{hypo}$  จากทุกเส้นทางดำเนินงานที่สร้างออกมาโดยเครื่องสร้างปริภูมิสถานะ โดยเริ่มจากสถานะเริ่มต้น  $s_0$  เมื่อพบสถานะสมมติฐานแล้ว ค้นหาสถานะถัดไปว่าเป็นสถานะข้อสรุป  $CS_{conclus}$  หรือไม่ หากพบว่าเป็นสถานะข้อสรุปบนทุกเส้นทางดำเนินงานที่พบสถานะสมมติฐานจะทำให้ผลลัพธ์ก็จะเป็นที่พอใจ ในทำนองเดียวกัน หากไม่พบสถานะสมมติฐานบนทุกเส้นทางดำเนินงานจะได้ผลที่พอใจเช่นเดียวกัน ในทางกลับกัน หากสถานะถัดไปไม่ใช่สถานะข้อสรุปในบางเส้นทางดำเนินงานผลลัพธ์ก็จะเป็นที่ไม่พอใจ

#### 4.4.4 การทดสอบคุณสมบัติสูตรแอลทีแอลประเภท Until

กระบวนการทดสอบสูตรแอลทีแอลประเภท Until เริ่มจากการอ่านสูตรแอลทีแอลแบบเฉพาะ คือ  $SLF_{until} = \Phi_{req} \cup \Psi_{ack}$  จากนั้น แปลงชื่อเพลสไปยังชื่อสถานะที่สอดคล้องกัน คือ  $CS_{req}$  เป็นสถานะที่ร้องขอและ  $CS_{ack}$  เป็นสถานะตอบกลับ สามารถแสดงเป็นแผนภาพกิจกรรมดังรูปที่ 4.7



รูปที่ 4.7 แผนภาพกิจกรรมการทวนสอบสูตรแอลที่แอลประเภท Until

จากรูปที่ 4.7 เครื่องมือการทวนสอบตรวจสอบว่าทุกสถานะตอบกลับ  $CS_{ack}$  เป็นสถานะเริ่มต้นหรือไม่ หากพบว่าทุกสถานะตอบกลับ  $CS_{ack}$  เป็นสถานะเริ่มต้นจะได้ผลลัพธ์เป็นที่พอใจ หากพบว่ามีบางสถานะไม่เป็นสถานะเริ่มต้น เครื่องมือการทวนสอบจะเริ่มต้นค้นหาว่าสถานะร้องขอ  $CS_{req}$  เป็นสถานะเริ่มต้นหรือไม่ หากพบว่าไม่เป็นสถานะเริ่มต้นจะส่งผลลัพธ์เป็นที่ไม่พอใจ แต่หากพบว่าเป็นทุกสถานะเป็นสถานะเริ่มต้น เครื่องมือการทวนสอบจะเริ่มสร้างปริภูมิสถานะออกมาโดยใช้

เครื่องสร้างปริภูมิสถานะ โดยจะทำการตรวจสอบว่าสถานะถัดไปเป็นสถานะตอบกลับ  $CS_{ack}$  หรือไม่ หากพบว่าสถานะถัดไปเป็นสถานะร้องขอ  $CS_{req}$  เครื่องมือการทวนสอบจะทำการค้นหาสถานะถัดไป ต่อเรื่อย ๆ จนกระทั่งพบว่าสถานะถัดไปไม่ใช่สถานะร้องขอ  $CS_{req}$  จากนั้น เครื่องมือการทวนสอบจะทำการค้นหาสถานะถัดไปว่าเป็นสถานะตอบกลับ  $CS_{ack}$  หรือไม่ หากพบว่า เป็นสถานะตอบกลับบนทุกเส้นทางดำเนินงานจะทำให้ผลลัพธ์ก็จะเป็นที่พอใจ แต่ในทางกลับกัน ในระหว่างการตรวจสอบ หากมีการวนลูปหรือพบสถานะสุดท้ายหรือตรวจสอบแล้วว่าสถานะถัดไปไม่ใช่สถานะตอบกลับ  $CS_{ack}$  จะทำให้ส่งผลลัพธ์เป็นที่ไม่พอใจ



## บทที่ 5

### การออกแบบสถาปัตยกรรมระบบและการพัฒนาระบบ

ในบทนี้ อธิบายฟังก์ชันการทำงานและสถาปัตยกรรมของเครื่องมือสร้างปฏิภูมิสถานะสำหรับโมเดลเช็กกิงแบบแอลทีแอลใช้แอลเอสทีเอ็มที่มีการจำแนกแบบหลายป้าย ซึ่งถูกเรียกว่า เครื่องสร้างปฏิภูมิสถานะ และเครื่องมือการทวนสอบสูตรแอลทีแอลแบบเฉพาะ ซึ่งถูกเรียกว่า เครื่องมือการทวนสอบ เมื่อได้เครื่องมือสร้างปฏิภูมิสถานะและเครื่องมือการทวนสอบแล้ว ผู้วิจัยได้สร้างเป็นระบบการสร้างปฏิภูมิสถานะและการทวนสอบคุณสมบัติที่ใช้งานส่วนของเครื่องมือสร้างปฏิภูมิสถานะและเครื่องมือการทวนสอบ

#### 5.1 การวิเคราะห์และออกแบบระบบ

ผู้วิจัยได้วิเคราะห์และออกแบบระบบโดยเริ่มจากการวิเคราะห์ขอบเขตและการทำงานของระบบ เพื่อให้ได้เป็นความต้องการเชิงฟังก์ชัน (Functional requirement) จากนั้น ออกแบบระบบโดยใช้แผนภาพที่แสดงให้เห็นถึงความสัมพันธ์ระหว่างผู้ใช้งานและการทำงานภายในระบบ

##### 5.1.1 ความต้องการเชิงฟังก์ชัน

จากการวิเคราะห์ขอบเขตและฟังก์ชันการทำงานหลักของระบบ ซึ่งมีทั้งหมด 13 รายการดังต่อไปนี้

- 1) **FNR-01:** เลือกไต่แรกทอรีของเครื่องมือสร้างปฏิภูมิสถานะ  
ผู้ใช้สามารถเรียกดูและเลือกไต่แรกทอรีของเครื่องมือสร้างปฏิภูมิสถานะ สำหรับระบบที่ผู้ใช้สนใจ
- 2) **FNR-02:** สร้างปฏิภูมิสถานะ  
เมื่อผู้ใช้ป้อนข้อมูลที่จำเป็นสำหรับการสร้างปฏิภูมิสถานะเรียบร้อยแล้ว ระบบดำเนินการสร้างปฏิภูมิสถานะ โดยพิจารณาจากข้อมูลที่ผู้ใช้ป้อนเข้ามา
- 3) **FNR-03:** ตรวจสอบความถูกต้องของข้อมูลนำเข้า  
เมื่อผู้ใช้ป้อนข้อมูลนำเข้า ระบบจะต้องตรวจสอบความถูกต้องของข้อมูลที่นำเข้ามา คือ
  - 3.1) ไต่แรกทอรีของเครื่องมือสร้างปฏิภูมิสถานะ
  - 3.2) เส้นทางดำเนินงานที่ถูกระบุโดยผู้ใช้
  - 3.3) สูตรแอลทีแอลที่ผู้ใช้กำหนด
  - 3.4) ชุดของสูตรแอลทีแอลแบบเฉพาะที่อยู่ในรูปแบบของไฟล์ “.csv”
- 4) **FNR-04:** แสดงปฏิภูมิสถานะที่สร้างขึ้น  
แสดงผลการสร้างปฏิภูมิสถานะที่ถูกสร้างขึ้น

- 5) **FNR-05:** นำออกปฏิภูมิสถานะที่สร้างขึ้น  
การออกรายงานการสร้างปฏิภูมิสถานะที่สร้างขึ้น ในรูปไฟล์นามสกุล “.csv”
- 6) **FNR-06:** ทวนสอบคุณสมบัติการทำงานของระบบ  
เมื่อรับข้อมูลนำเข้าจากผู้ใช้แล้ว ระบบทวนสอบสูตรแอลทีแอลทีที่ได้รับเข้ามา
- 7) **FNR-07:** กรอกข้อมูลนำเข้าสูตรแอลทีแอลแบบเฉพาะ  
กรอกข้อมูลนำเข้าจากผู้ใช้ที่เป็นสูตรแอลทีแอลแบบเฉพาะ
- 8) **FNR-08:** เลือกนำเข้าชุดข้อมูลนำเข้าสูตรแอลทีแอลแบบเฉพาะ  
ผู้ใช้สามารถเลือกชุดข้อมูลนำเข้าสูตรแอลทีแอลแบบเฉพาะที่อยู่ในรูปไฟล์นามสกุล “.csv”
- 9) **FNR-09:** แสดงผลการทวนสอบคุณสมบัติการทำงานของระบบ  
การแสดงผลจะสามารถแสดงได้ 2 แบบ คือ
  - 9.1) หากผู้ใช้เป็นผู้ป้อนสูตรแอลทีแอล ระบบแสดงผลการทวนสอบผ่านหน้าจอ
  - 9.2) หากผู้ใช้นำเข้าชุดของสูตรแอลทีแอลแบบเฉพาะที่อยู่ในรูปไฟล์นามสกุล “.csv” ระบบจะออกรายการผลการทวนสอบที่อยู่ในรูปไฟล์นามสกุล “.csv”
- 10) **FNR-10:** ลบผลการทวนสอบ  
ระบบสามารถให้ผู้ใช้ลบผลการทวนสอบที่แสดงในหน้าจอได้
- 11) **FNR-11:** แสดงจำนวนเส้นทางดำเนินงานทั้งหมดที่ถูกสร้างออกมา  
เมื่อระบบสร้างปฏิภูมิสถานะเรียบร้อยแล้ว จำนวนของเส้นทางดำเนินงานจะต้องถูกแสดงที่หน้าจอ
- 12) **FNR-12:** แสดงตัวอย่างสูตรแอลทีแอลแบบเฉพาะ ตามที่ผู้ใช้เลือก  
เมื่อผู้ใช้เลือกว่าต้องการทวนสอบคุณสมบัติใด ระบบจะต้องแสดงตัวอย่างวิธีการเขียนสูตรแอลทีแอลที่ถูกต้องที่หน้าจอ เพื่อให้ผู้ใช้สามารถเขียนสูตรแอลทีแอลที่ถูกต้อง
- 13) **FNR-13:** แสดงสถานะการทำงานของระบบ  
ระบบจะต้องแสดงสถานะการทำงานของระบบ เช่น ระบบพร้อมใช้งาน ระบบกำลังทำสร้างหรือกำลังทวนสอบ เป็นต้น

### 5.1.2 การออกแบบระบบ

จากความต้องการเชิงฟังก์ชันจากหัวข้อก่อนหน้า ผู้วิจัยได้ออกแบบระบบด้วยแผนภาพยูสเคส (Use case diagram) เพื่อแสดงการทำงานและความสัมพันธ์ระหว่างผู้ใช้งานและระบบงานย่อย ดังรูปที่ 5.1 และคำอธิบายยูสเคสแสดงในภาคผนวก ก



รูปที่ 5.1 แผนภาพยูสเคสของระบบการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติ

จากรูปที่ 5.1 สามารถสรุป ยูสเคสทั้งหมดของระบบของระบบการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติ โดยใช้เครื่องสร้างปริภูมิสถานะที่ใช้วิธีการผสมผสานแอลเอสทีเอ็มที่สามารถจัดการกับการจำแนกหลายป้าย ได้ทั้งหมด 13 ยูสเคส สามารถแสดงรายละเอียดได้ในตารางที่ 5.1

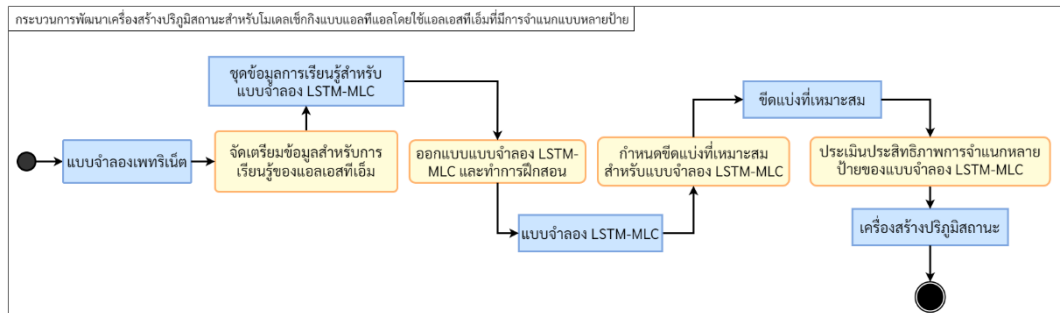
ตารางที่ 5.1 รายการแสดงยูสเคสทั้งหมด

รหัสยูสเคส	ชื่อยูสเคส	คำอธิบาย
UC01	Browse state space generator directory	อนุญาตให้ผู้ใช้เรียกดูและเลือกไดเรกทอรีของเครื่องสร้างปริภูมิสถานะ สำหรับระบบที่สนใจ
UC02	Generate state space	สร้างปริภูมิสถานะของระบบที่สนใจ
UC03	Validate user input	ตรวจสอบการป้อนข้อมูลของผู้ใช้เพื่อความปลอดภัยและครบถ้วน
UC04	Show generated state space	แสดงปริภูมิสถานะที่สร้างขึ้น โดยอยู่ในรูปแบบของเส้นทางดำเนินงาน
UC05	Export generated state space	ส่งออกปริภูมิสถานะที่สร้างขึ้นที่อยู่ในรูปแบบของเส้นทางดำเนินงาน
UC06	Verify LTL property	การทวนสอบคุณสมบัติการทำงานของระบบที่สนใจ
UC07	Fill LTL formula	อนุญาตให้ผู้ใช้ป้อนสูตรแอลทีแอลด้วยตัวเอง
UC08	Import LTL formula file	อนุญาตให้ผู้ใช้นำเข้าไฟล์สำหรับสูตรแอลทีแอล
UC09	Show verification result	แสดงผลลัพธ์การทวนสอบคุณสมบัติการทำงาน ของระบบ
UC10	Delete verification result	อนุญาตให้ผู้ใช้สามารถลบผลการทวนสอบได้
UC11	Show the total number of the generated execution paths	แสดงจำนวนเส้นทางดำเนินงานทั้งหมด
UC12	Show example of specific LTL formulas	แสดงตัวอย่างสูตรแอลทีแอลแบบเฉพาะ
UC13	Show status of the system	แสดงสถานะการทำงาน of ระบบ

## 5.2 การออกแบบสถาปัตยกรรม

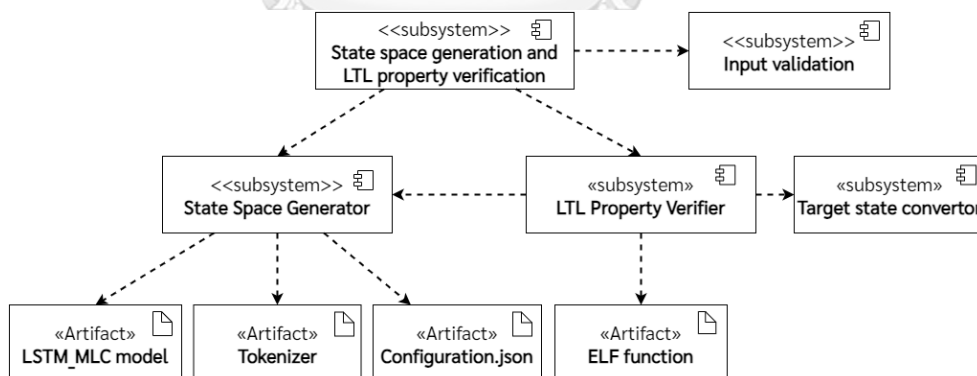
จากขั้นตอนการพัฒนาเครื่องสร้างปริภูมิสถานะที่ได้อธิบายไว้ในบทที่ 3 สามารถแสดงเป็นแผนภาพกิจกรรมภาพรวมของกระบวนการพัฒนาการสร้างเครื่องสร้างปริภูมิสถานะ สามารถแสดงได้

รูปที่ 5.2 เริ่มจากการออกแบบและสร้างแบบจำลอง LSTM-MLC จากนั้นนำแบบจำลองมาฝึกอบรมด้วยชุดข้อมูลที่จัดเตรียมไว้ก่อนหน้านี้ เมื่อได้แบบจำลองที่มีประสิทธิภาพในการเรียนรู้รูปแบบจำลอง LSTM-MLC ถูกนำมาหาขีดแบ่งที่เหมาะสม เพื่อนำไปประเมินประสิทธิภาพของแบบจำลอง LSTM-MLC เมื่อประเมินแล้วพบว่าแบบจำลองมีประสิทธิภาพในการทำงาน ก็จะได้เป็นเครื่องสร้างปฏิภูมิสถานะ



รูปที่ 5.2 แผนภาพกิจกรรมการพัฒนาการสร้างเครื่องสร้างปฏิภูมิสถานะ

เมื่อได้เครื่องสร้างปฏิภูมิสถานะที่มีประสิทธิภาพในการทำงานเรียบร้อยแล้ว ผู้วิจัยได้ออกแบบออกแบบสถาปัตยกรรมของระบบการสร้างปฏิภูมิสถานะและการทวนสอบคุณสมบัติ โดยแสดงเป็นแผนภาพระบบย่อย (Subsystem diagram) เพื่อแสดงให้เห็นระบบงานย่อยต่างๆ ภายในระบบการสร้างปฏิภูมิสถานะและการทวนสอบคุณสมบัติของระบบ แสดงในรูปที่ 5.3



รูปที่ 5.3 แผนภาพระบบย่อยของระบบการสร้างปฏิภูมิสถานะและการทวนสอบคุณสมบัติของระบบ

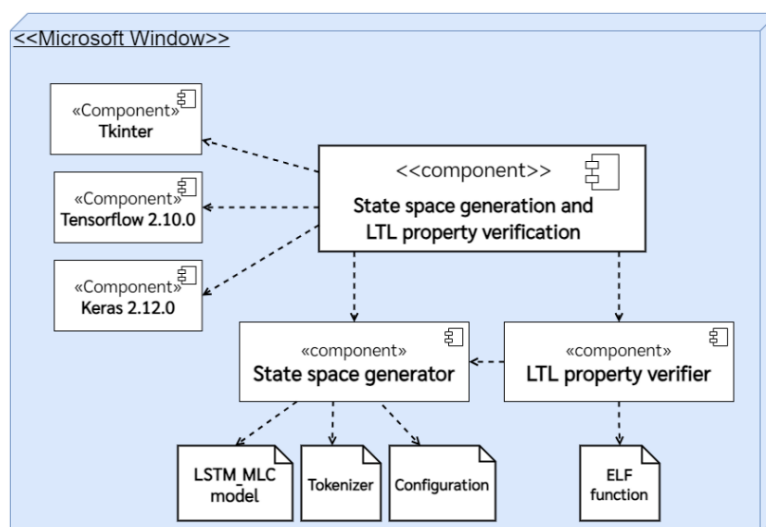
จากรูปที่ 5.3 พบว่ามีระบบย่อย 5 ระบบย่อย ดังต่อไปนี้

- 1) การสร้างปฏิภูมิสถานะและการทวนสอบคุณสมบัติแอลทีแอล (State space generation and LTL property verification) เป็นระบบย่อยหลักที่มีการเรียกใช้ระบบย่อยต่างๆ ซึ่งทำให้ผู้ใช้สามารถสร้างปฏิภูมิสถานะและทวนสอบคุณสมบัติการทำงานได้



- 2) การตรวจสอบข้อมูลนำเข้า (Input validation) เป็นระบบย่อยที่มีหน้าที่ ตรวจสอบความถูกต้องของข้อมูลนำเข้าจากผู้ใช้งานเข้ามาในระบบ
- 3) เครื่องสร้างปริภูมิสถานะ (State space generator) เป็นระบบย่อยที่มีหน้าที่ สร้างปริภูมิสถานะจากข้อมูลนำเข้าที่ผู้ใช้งานกำหนด ซึ่งเรียกใช้สิ่งประดิษฐ์ (Artifact) ได้แก่
  - 3.1) แบบจำลอง LSTM-MLC ที่เป็นแบบจำลองที่ประสิทธิภาพในการเรียนรู้ข้อมูลการฝึกสอนและสามารถนำไปใช้งานได้
  - 3.2) โทเคนไนเซอร์ (Tokenizer) เป็นวัตถุ (Object) ที่มีหน้าที่เก็บรวบรวมพจนานุกรมโทเคนสำหรับระบบที่สนใจ ใช้สำหรับแปลงข้อมูลดิบไปเป็นข้อมูลที่อยู่ในรูปแบบที่แบบจำลอง LSTM-MLC สามารถนำไปใช้งานได้
  - 3.3) ไฟล์การกำหนดค่า (Configuration file) เก็บรวบรวมองค์ประกอบที่จำเป็นในการสร้างปริภูมิสถานะ เช่น ซิตแบ่งที่เหมาะสม เป็นต้น
- 4) เครื่องมือการทวนสอบคุณสมบัติแอลทีแอล (LTL property verifier) เป็นระบบย่อยมีหน้าที่ ทวนสอบคุณสมบัติแอลทีแอลจากข้อมูลที่ผู้ใช้งานกำหนดโดยการเรียกใช้งานเครื่องสร้างปริภูมิสถานะและเรียกใช้สิ่งประดิษฐ์ ได้แก่
  - 4.1) ฟังก์ชันอีแอลเอฟ (ELF function) เป็นวัตถุที่มีหน้าที่แปลงชื่อของเพลสไปยังสถานะที่สอดคล้องกัน
- 5) ตัวแปลงสถานะเป้าหมาย (Target state convertor) เป็นระบบย่อยที่ถูกเรียกใช้จากเครื่องมือการทวนสอบเมื่อผู้ใช้งานสูตรแอลทีแอลเข้ามา โดยทำหน้าที่แปลงชื่อเพลสไปยังสถานะที่สอดคล้องกันสำหรับแต่ละเพลส เพื่อให้ได้สถานะเป้าหมายทั้งหมดที่จะถูกนำไปใช้ในกระบวนการทวนสอบคุณสมบัติของระบบสำหรับงานวิจัยนี้

จากนั้น ผู้วิจัยได้ออกแบบแผนภาพสถาปัตยกรรมของระบบที่แสดงโดยใช้แผนภาพดีพลอยเมนต์ (Deployment diagram) สำหรับการติดตั้งโปรแกรมไพธอน (Python) เพื่อแสดงให้เห็นลักษณะทางกายภาพของระบบการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติ ดังรูปที่ 5.4



รูปที่ 5.4 แผนภาพดีพลอยเมนต์ของระบบการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติ

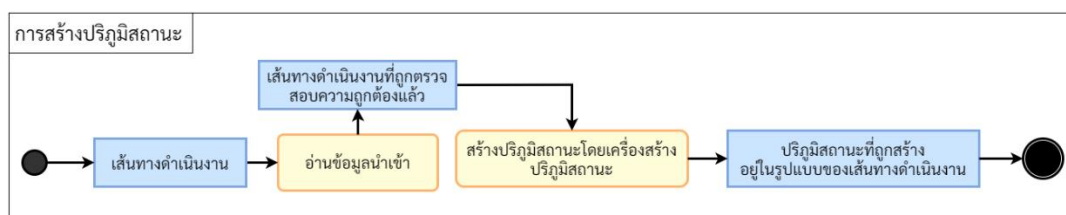
จากรูปที่ 5.4 แสดงให้เห็นถึงลักษณะทางกายภาพของระบบการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติที่พัฒนาโดยใช้ไพธอน 3 (Python3) [63] ที่ติดตั้งบนหน้าต่างไมโครซอฟต์ที่เชื่อมต่อประสานกับผู้ใช้เพื่อให้สามารถสร้างปริภูมิสถานะหรือการทวนสอบคุณสมบัติของระบบได้ โดยมีองค์ประกอบหลัก ดังนี้

- 1) การสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติแอลทีแอล (State space generation and LTL property verification) คือส่วนประกอบสำหรับการทำงานหลักที่พัฒนาเป็นหน้าต่างแอปพลิเคชันที่ใช้ต่อประสานกับผู้ใช้งาน ซึ่งถูกสร้างมาจากระบบย่อยของการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติแอลทีแอล (State space generation and LTL property verification subsystem) และระบบย่อยสำหรับการตรวจสอบข้อมูลนำเข้า (Input validation subsystem) เมื่อผู้ใช้งานต้องการสร้างปริภูมิสถานะหรือทวนสอบคุณสมบัติการทำงานของระบบ ผู้ใช้สามารถเรียกใช้การทำงานผ่านส่วนประกอบหลักนี้ นอกจากนี้ ยังมีส่วนประกอบภายนอกที่ถูกเรียกใช้เพื่อแสดงผลการสร้างปริภูมิสถานะหรือผลการทวนสอบได้ ซึ่งจำเป็นต้องมีส่วนประกอบที่ถูกติดตั้ง ดังต่อไปนี้
  - 1.1) Tkinter [64] ไลบรารีสำหรับพัฒนาหน้าต่างแอปพลิเคชันสำหรับภาษาไพธอน
  - 1.2) Tensorflow [65] ไลบรารีสำหรับการพัฒนาเครื่องสร้างปริภูมิสถานะด้วยการเรียนรู้ของแอลเอสทีเอ็ม เวอร์ชัน 2.10.0
  - 1.3) Kears [66] ไลบรารีสำหรับภาษาไพธอนที่ใช้ในการพัฒนาเครื่องสร้างปริภูมิสถานะด้วยการเรียนรู้ของแอลเอสทีเอ็ม เวอร์ชัน 2.12.0

- 2) เครื่องสร้างปริภูมิสถานะ (State space generator) เป็นส่วนประกอบของการสร้างปริภูมิสถานะ ซึ่งถูกสร้างมาจากระบบย่อยของเครื่องสร้างปริภูมิสถานะ (State space generator subsystem) ที่มีการเรียกใช้ 3 สิ่งประดิษฐ์ ที่ถูกพัฒนามาตามขั้นตอนในบทที่ 3 ดังนี้
  - 2.1) แบบจำลอง LSTM\_MLC ซึ่งเป็นแบบจำลอง LSTM-MLC ที่ผ่านการประเมินประสิทธิภาพการเรียนรู้เรียบร้อยแล้ว
  - 2.2) โทเคนไนเซอร์ (Tokenizer) วัตถุที่ถูกสร้างขึ้นมาจากบทที่ 3 สำหรับใช้ในการแปลงข้อมูลนำเข้าเพื่อให้แบบจำลอง LSTM-MLC สามารถนำไปใช้ในการทำนายผลได้
  - 2.3) ไฟล์กำหนดค่า (Configuration) ไฟล์เอกสารที่มีการกำหนดค่าที่รวบรวมข้อมูลที่จำเป็นที่ต้องใช้ในการทำนาย เช่น ขีดแบ่งที่เหมาะสม เป็นต้น
- 3) เครื่องมือการทวนสอบคุณสมบัติแอลทีแอล (LTL property verifier) เป็นส่วนประกอบของการตรวจสอบคุณสมบัติของระบบ ซึ่งถูกสร้างมาจากระบบย่อยของเครื่องมือการทวนสอบคุณสมบัติแอลทีแอล (LTL property verify subsystem) และระบบย่อยของตัวแปลงสถานะเป้าหมาย (Target state convertor subsystem) ที่มีการเรียกใช้ฟังก์ชันอีแอลเอส (ELF function) เพื่อทำการแปลงชื่อของเพลสไปยังสถานะที่สอดคล้อง โดยการเรียกใช้ระบบย่อยของตัวแปลงสถานะเป้าหมาย (Target state convertor subsystem)

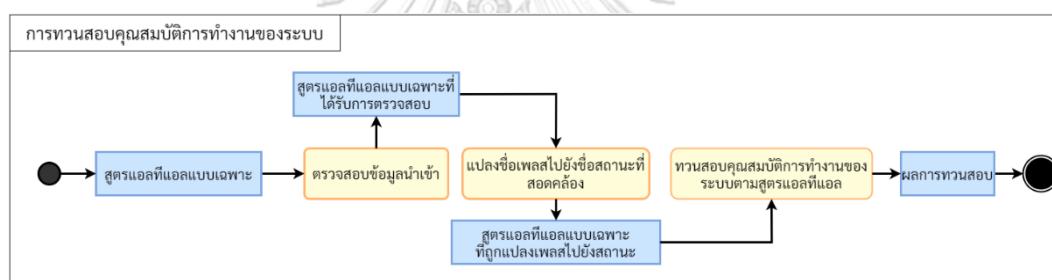
จากการออกแบบสถาปัตยกรรมของระบบการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติแอลทีแอล ซึ่งสามารถแบ่ง 2 ส่วน คือ การสร้างปริภูมิสถานะโดยใช้เครื่องสร้างปริภูมิสถานะ แสดงได้ในรูปที่ 5.5 และการทวนสอบคุณสมบัติการทำงานของระบบ โดยการกำหนดสูตรแอลทีแอลแบบเฉพาะโดยใช้เครื่องมือการทวนสอบที่อธิบายไว้ในบทที่ 4 แสดงได้ในรูปที่ 5.6

จากแผนภาพกิจกรรมที่แสดงในรูปที่ 5.5 การทำงานเริ่มต้นเมื่ออ่านข้อมูลนำเข้าที่ถูกป้อนจากผู้ใช้งาน จากนั้นเริ่มสร้างปริภูมิสถานะตามจำนวนความยาวของเส้นทางดำเนินงานที่ผู้ต้องการ โดยเรียกใช้เครื่องสร้างปริภูมิสถานะถูกสร้างไว้ สุดท้ายปริภูมิสถานะที่สร้างถูกแสดงออกมาในรูปแบบของเส้นทางดำเนินงาน



รูปที่ 5.5 แผนภาพกิจกรรมการสร้างปริภูมิสถานะ

จากแผนภาพกิจกรรมที่แสดงในรูปที่ 5.6 การทำงานเริ่มต้นเมื่อ รับข้อมูลนำเข้าเป็นสูตรแอลที่แอลแบบเฉพาะ จากนั้น ทำการตรวจสอบข้อมูลนำเข้าและแปลงชื่อเพลสไปยังสถานะที่สอดคล้องกันเพื่อนำไปใช้ในการทวนสอบ ในระหว่างการทวนสอบเครื่องสร้างปริภูมิสถานะถูกเรียกใช้งานเพื่อสร้างเส้นทางดำเนินงานโดยเริ่มจากสถานะเริ่มต้น คือ s0 ไปจนถึงความลึกที่ผู้ใช้กำหนด เมื่อทำการทวนสอบเสร็จแล้ว ผลการทวนสอบถูกแสดงในหน้าจอ หากกำหนดความลึกจำนวนมากอาจทำให้ใช้เวลาในการทวนสอบนาน



รูปที่ 5.6 แผนภาพกิจกรรมการทวนสอบคุณสมบัติการทำงานของระบบ

### 5.3 การพัฒนาระบบ

สำหรับการพัฒนาระบบ ผู้วิจัยได้แสดงสภาพแวดล้อมทางด้านฮาร์ดแวร์และซอฟต์แวร์ที่จะใช้ในการพัฒนาระบบสำหรับงานวิจัยนี้ ประกอบด้วย

#### 1) ด้านฮาร์ดแวร์

- 1.1) หน่วยประมวลผล อินเทลเพนเทียมที่เอ็ม 2.40 กิกะเฮิร์ต (Core i7 TM 2.40 GHz.)
- 1.2) หน่วยความจำ (RAM) 8 กิกะไบต์ (8.0 GB)
- 1.3) ฮาร์ดดิสก์ (Hard disk) 1 เทระไบต์ (1 TB)

#### 2) ด้านซอฟต์แวร์

- 2.1) เครื่องมือพัฒนาโปรแกรมวิซวลสตูดิโอโค้ด เวอร์ชัน 1.82.1
- 2.2) เครื่องมือมินิคอนดา 3 (Miniconda3 tool)

### 2.3) เครื่องเสมือน (Virtual machine) จากแล็บอาจารย์บุญเสริม

หลังจากเตรียมเครื่องมือสำหรับการพัฒนาระบบแล้ว ผู้วิจัยได้ทำการติดตั้งเครื่องมือที่ใช้ในการพัฒนาระบบ ตามลำดับดังต่อไปนี้

- 1) ติดตั้งโปรแกรมวิชวลสตูดิโอโค้ดลงบนระบบปฏิบัติการวินโดวส์ 11
- 2) ติดตั้งเครื่องมือมินิคอนดา 3 และจำเป็นต้องติดตั้ง Python3 Tkinter Tensorflow 2.10.0 และ Kears 2.12.0

หลังจากติดตั้งเครื่องมือที่ใช้ในการพัฒนาระบบเรียบร้อยแล้ว ผู้วิจัยได้ออกแบบและพัฒนาส่วนต่อประสานกับผู้ใช้ให้สอดคล้องกับการทำงานของระบบที่ตรงกับความต้องการเชิงฟังก์ชัน ซึ่งรายละเอียดของส่วนต่อประสานผู้ใช้อธิบายในภาคผนวก ข



## บทที่ 6

### การทดสอบเครื่องสร้างปริภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอตทิลโดยใช้อัลเอสทีเอ็มที่มี การจำแนกแบบหลายป้าย

บทนี้อธิบายเกี่ยวกับการทดสอบเครื่องสร้างปริภูมิสถานะสำหรับโมเดลเชิงกึ่งแบบแอตทิลโดยใช้อัลเอสทีเอ็มที่มีการจำแนกแบบหลายป้าย โดยแบ่งเป็น 3 ส่วน ส่วนแรกเป็นการออกแบบกรณีศึกษา ส่วนที่สองเป็นการออกแบบวิธีการทดสอบและส่วนสุดท้ายเป็นการทดสอบและรายงานผลการทดสอบตามกรณีทดสอบที่ได้ออกแบบไว้

#### 6.1 การออกแบบกรณีศึกษา

ในส่วนแรก ผู้วิจัยได้จัดเตรียมกรณีศึกษา (Case study) ที่ใช้ในการทดสอบ ซึ่งเป็นแบบจำลองเพทรีเน็ตที่จำลองการทำงานของระบบที่มีจำนวนสถานะและลักษณะของกราฟปริภูมิสถานะที่แตกต่างกัน ซึ่งมีทั้งหมด 4 กรณีศึกษา ดังต่อไปนี้

**กรณีศึกษาที่ 1** ระบบเครื่องรับจ่ายเงินอัตโนมัติหรือเอทีเอ็ม (Automated teller machine system หรือ ATM) เป็นระบบที่มีที่สิ้นสุด (Finite system) ผู้วิจัยออกแบบขึ้นมาเพื่อใช้ในการแสดงการเข้าถึงมาร์กกิงทั้งหมดและกราฟปริภูมิสถานะของระบบ เนื่องจาก ระบบนี้สามารถอธิบายให้เข้าใจได้ง่ายกว่าระบบอื่น ซึ่งวิธีการได้มาซึ่งมาร์กกิงทั้งหมดและกราฟปริภูมิสถานะของทุกกรณีศึกษาจะใช้วิธีเดียวกัน โดยระบบนี้มีฟังก์ชันการทำงานหลัก คือ การถอนเงิน การฝากเงินและการจ่ายบิล

**กรณีศึกษาที่ 2** ระบบการรับประทานอาหารของนักปราชญ์ 9 คนหรือดีพีเอส (Dinner 9 philosophers system หรือ DPS) [67] เป็นระบบที่ไม่มีที่สิ้นสุด (Infinite system) ที่จำลองมาจากปัญหาคลาสสิกที่มักถูกใช้เป็นตัวช่วยในการออกแบบอัลกอริทึมแบบพร้อมกันเพื่อแสดงปัญหาการชิงโครโนส์ โดยที่นักปราชญ์ 9 คนนั่งแบบล้อมกันเป็นวงกลม ซึ่งนักปราชญ์มีความสามารถในการคิดและกิน สลับกันเท่านั้น โดยแต่ละคนจะอยู่ในสถานะที่กำลังคิดและมีส้อมทั้งทางซ้ายและขวา เมื่อมีนักปราชญ์ต้องการรับประทานอาหาร นักปราชญ์จะต้องมีส้อมทางซ้ายและขวา หากไม่มีส้อมให้พร้อมใช้งาน หมายความว่า นักปราชญ์ที่นั่งติดกันกำลังรับประทานอาหาร ทำให้นักปราชญ์คนดังกล่าวจะต้องรอไปจนกว่าส้อมทั้งทางซ้ายและขวาพร้อมใช้งานพร้อมกัน

**กรณีศึกษาที่ 3** ระบบสัญญาณไฟจราจรสี่ด้านหรือเอฟทีแอล (Four traffics lights system หรือ FTL) เป็นระบบที่ไม่มีที่สิ้นสุด โดยการทำงานของแต่ละเสาไฟจราจรทำงานเป็นลำดับแบบพร้อม

กัน โดยมีสัญญาณไฟมีทั้งหมด 3 สี คือ ไฟแดง ไฟเขียวและไฟเหลือง ตามลำดับ โดยเริ่มจากเสาไฟจราจรต้นที่ 1 2 3 และ 4 ตามลำดับ

**กรณีศึกษาที่ 4** ระบบหน่วยความจำดาวเทียมแบบหรือเอสทีเอ็ม (Satellite memory system หรือ STM) [68] เป็นระบบที่ไม่มีที่สิ้นสุด ที่มาจากการงานประกวด Model Checking Contest 2020 ซึ่งเป็นเวอร์ชันดัดแปลงจากเวอร์ชันดั้งเดิม โดยการปรับเปลี่ยนจำนวนโทเคนให้มีความหลากหลาย

อย่างไรก็ดี วิธีการมาร์กกิงทั้งหมดของทุกกรณีศึกษาเป็นวิธีการที่ใช้ทั่วไป ดังนั้น มีเพียงกรณีศึกษาที่ 1 เท่านั้น ที่แสดงตัวอย่างของมาร์กกิงทั้งหมดและกราฟปริภูมิสถานะของกรณีศึกษาที่ 1 สามารถแสดงกราฟปริภูมิของทั้งระบบได้ ในขณะที่ กรณีศึกษาที่ 2 ถึง 4 กราฟปริภูมิสถานะที่แสดงขึ้นมาเป็นเพียงส่วนหนึ่งเท่านั้น เพื่อให้เห็นโครงสร้างของกราฟ ซึ่งกราฟปริภูมิสถานะทั้งหมดของกรณีศึกษาที่ 2 และ 3 ถูกแสดงในภาคผนวก ค โดยจำนวนสถานะของแต่ละกรณีศึกษาสามารถสรุปได้ในตารางที่ 6.1

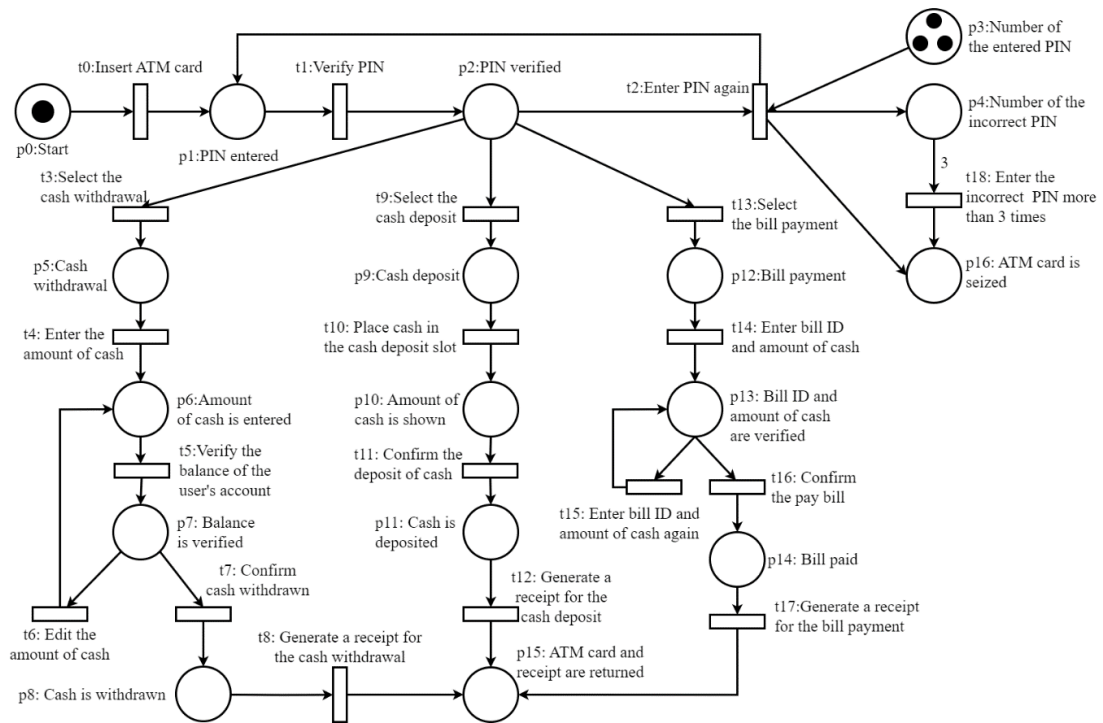
ตารางที่ 6.1 ลักษณะของกราฟปริภูมิสถานะของแต่ละกรณีศึกษา

กรณีศึกษา	จำนวนสถานะ	จำนวนชั้น	ลักษณะกราฟปริภูมิสถานะ
1. ระบบเครื่องรับจ่ายเงินอัตโนมัติ	54	12	ขนาดเล็กมีการวนซ้ำแต่ไม่ซับซ้อน
2. ระบบการรับประทานอาหารของนักปราชญ์ 9 คน	77	68	ขนาดใหญ่มีความซับซ้อนและการวนซ้ำ
3. ระบบสัญญาณไฟจราจรสี่ต้น	325	81	ขนาดใหญ่มีความซับซ้อน มีการวนซ้ำและขยายเป็นแนวกว้าง
4. ระบบหน่วยความจำดาวเทียม	17,326	8,782	ขนาดใหญ่มีความซับซ้อน มีการวนซ้ำและขยายเป็นแนวลึก

#### 6.1.1 กรณีศึกษาที่ 1 แบบจำลองเพทรีเน็ตสำหรับระบบเครื่องรับจ่ายเงินอัตโนมัติ

แบบจำลองเพทรีเน็ตสำหรับระบบเครื่องรับจ่ายเงินอัตโนมัติ แสดงในรูปที่ 6.1 ซึ่งมีความสามารถในการถอนเงิน ผ่าเงินและจ่ายบิล โดยรองรับผู้ใช้ครั้งละ 1 คนเท่านั้น เริ่มการทำงาน

เมื่อผู้ใช้บัตร ATM และจากนั้นกด PIN ซึ่งผู้ใช้สามารถใส่ PIN ผิดได้ไม่เกิน 3 ครั้ง หากผู้ใช้ใส่ PIN ผิดเกิน 3 ครั้ง บัตร ATM จะถูกยึด



รูปที่ 6.1 แบบจำลองเพทรีเน็ตสำหรับระบบเครื่องรับจ่ายเงินอัตโนมัติ

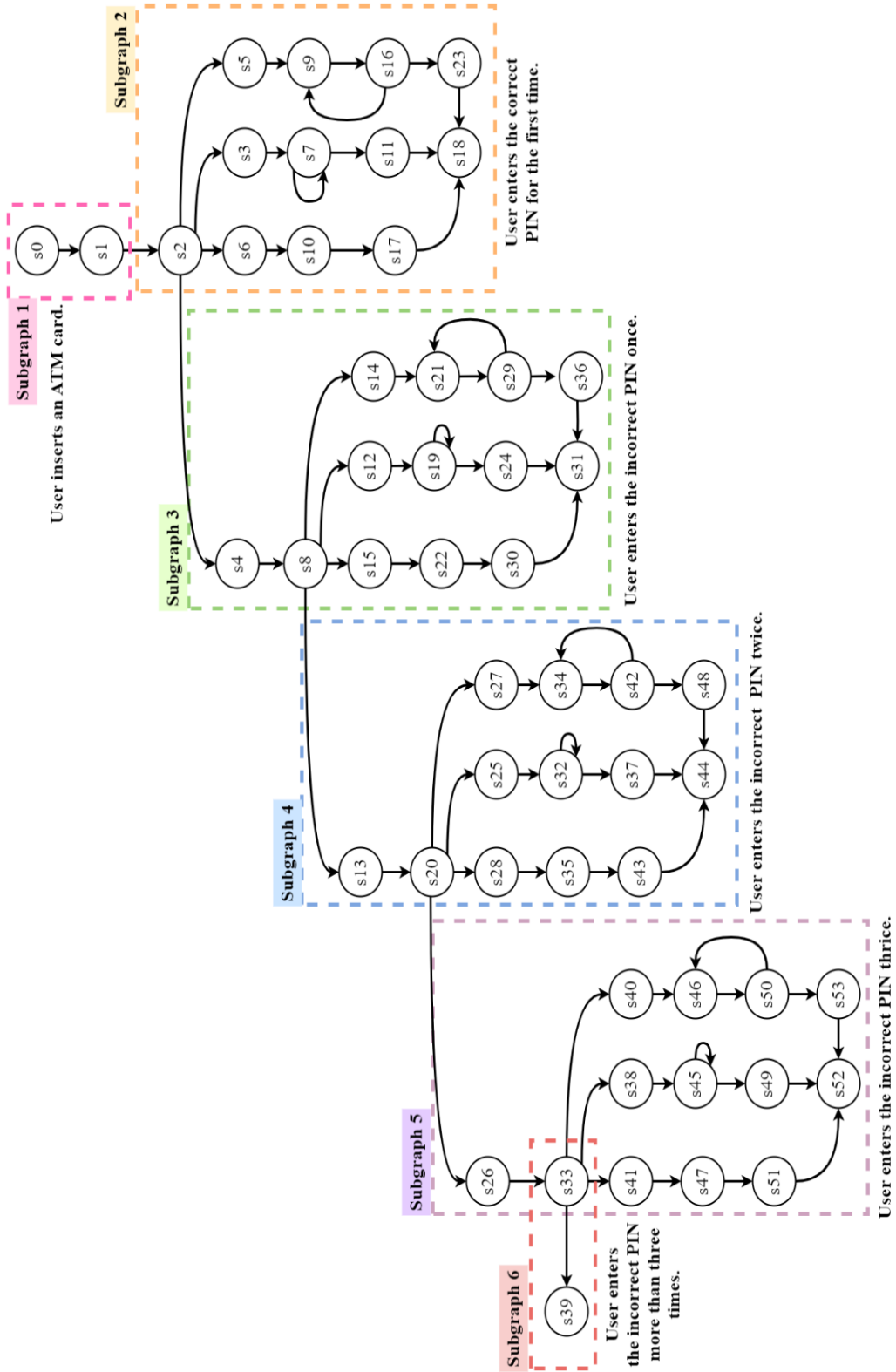
เมื่อได้ออกแบบจำลองเพทรีเน็ตแล้ว โดยรายชื่อเพลสและคำอธิบายแสดงในตารางที่ 6.2 จากนั้น ผู้วิจัยได้ทำการเข้าถึงมาร์กกิงทั้งหมดของระบบมีทั้งหมด 54 มาร์กกิงหรือ 54 สถานะสามารถแสดงได้ดังรูปที่ 6.2

ตารางที่ 6.2 รายชื่อเพลสและคำอธิบายของกรณีศึกษาที่ 1 ระบบเครื่องรับจ่ายเงินอัตโนมัติ

ชื่อเพลส	คำอธิบาย
p0: Start	ผู้ใช้เริ่มต้นใช้ระบบเครื่องกดเงินอัตโนมัติ
P1: EnteRedPIN	PIN ถูกป้อนเข้ามาในระบบ
p2: VerifiedPIN	PIN ได้รับการตรวจสอบความถูกต้อง
p3: NumEnteRedPIN	จำนวนครั้งที่ผู้ใช้ป้อน PIN
p4: NumIncorrectPIN	จำนวน PIN ที่ไม่ถูกต้อง
p5: CashWithdrawal	การถอนเงินสด
p6: EnteRedAmountCash	ป้อนจำนวนเงินสดที่ต้องการถอน
p7: VerifiedBalance	ยอดเงินที่ได้รับการตรวจสอบแล้ว



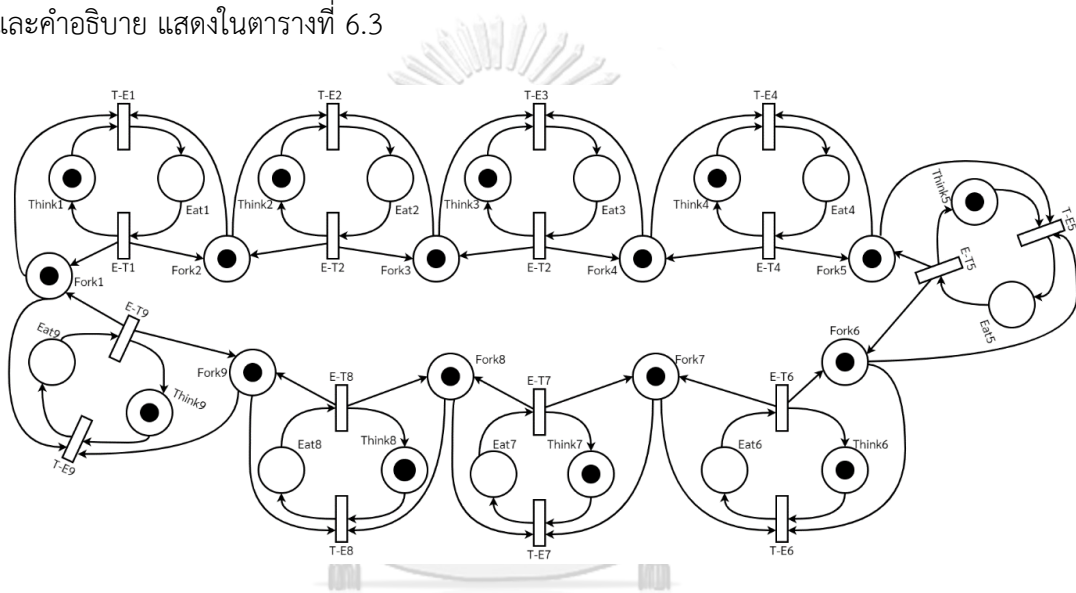




รูปที่ 6.3 กราฟประวิติสถานะของระบบเครื่องรับจ่ายเงินอัตโนมัติ

## 6.1.2 กรณีศึกษาที่ 2 แบบจำลองเพทรีเน็ตสำหรับระบบการรับประทานอาหารของนักปราชญ์ 9 คน

แบบจำลองเพทรีเน็ตสำหรับระบบการรับประทานอาหารของนักปราชญ์ 9 คน แสดงในรูปที่ 6.4 โดยนักปราชญ์ทุกคนอยู่ในสถานะสถานะการณืเริ่มต้น คือ ทุกคนกำลังคิดและล้อมทั้งหมด 10 คัน พร้อมใช้งาน ซึ่งแต่ละคนสามารถคิดและกินได้เท่านั้น โดยการกินมีเงื่อนไขคือ นักปราชญ์คนนั้นจำเป็นต้องมีส้อมพร้อมใช้งานอยู่ทั้ง 2 ข้าง เมื่อรับประทานอาหารเสร็จเรียบร้อยแล้ว นักปราชญ์จะกลับไปคิดและวางส้อมลง เพื่อให้ให้นักปราชญ์ท่านอื่นสามารถรับประทานอาหารได้ แต่หากไม่มีส้อมว่าง นักปราชญ์ท่านนั้นจำเป็นต้องรอไปเรื่อย ๆ จนกว่ามีส้อมพร้อมใช้งานทั้งสองข้าง ซึ่งรายชื่อเพลสและคำอธิบาย แสดงในตารางที่ 6.3



รูปที่ 6.4 แบบจำลองเพทรีเน็ตสำหรับระบบการรับประทานอาหารของนักปราชญ์ 9 คน

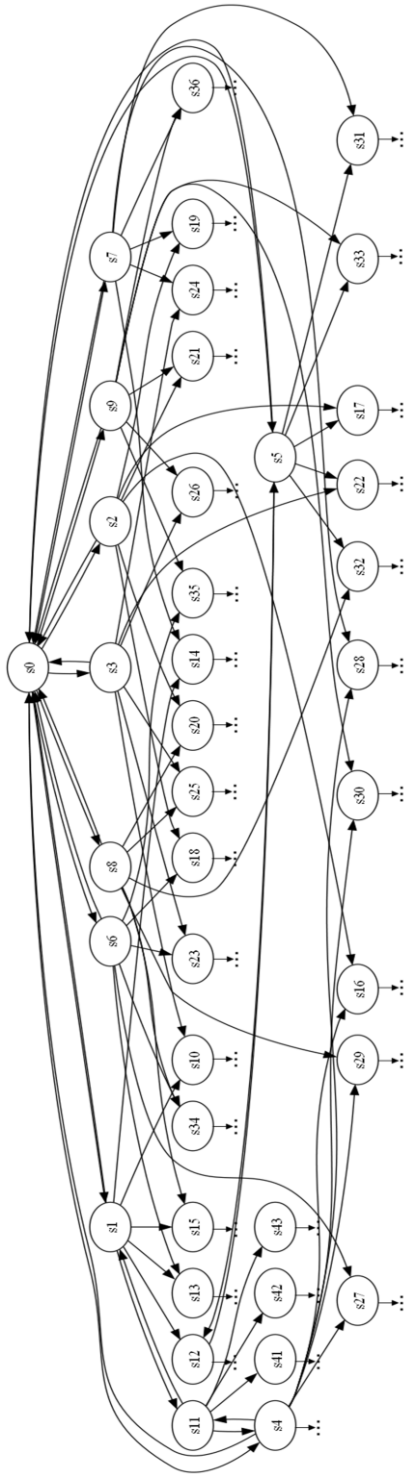
ตารางที่ 6.3 รายชื่อเพลสและคำอธิบายของกรณีศึกษาที่ 2 ระบบการรับประทานอาหารของนักปราชญ์ 9 คน

ชื่อเพลส	คำอธิบาย
Thinking1	นักปราชญ์คนที่ 1 กำลังคิด
Thinking2	นักปราชญ์คนที่ 2 กำลังคิด
Thinking3	นักปราชญ์คนที่ 3 กำลังคิด
Thinking4	นักปราชญ์คนที่ 4 กำลังคิด
Thinking5	นักปราชญ์คนที่ 5 กำลังคิด
Thinking6	นักปราชญ์คนที่ 6 กำลังคิด
Thinking7	นักปราชญ์คนที่ 7 กำลังคิด

ตารางที่ 6.3 รายชื่อเพลสและคำอธิบายของกรณีศึกษาที่ 2 ระบบการรับประทานอาหารของนักปราชญ์ 9 คน (ต่อ)

ชื่อเพลส	คำอธิบาย
Thinking8	นักปราชญ์คนที่ 8 กำลังคิด
Thinking9	นักปราชญ์คนที่ 9 กำลังคิด
Eating1	นักปราชญ์คนที่ 1 กำลังกิน
Eating2	นักปราชญ์คนที่ 2 กำลังกิน
Eating3	นักปราชญ์คนที่ 3 กำลังกิน
Eating4	นักปราชญ์คนที่ 4 กำลังกิน
Eating5	นักปราชญ์คนที่ 5 กำลังกิน
Eating6	นักปราชญ์คนที่ 6 กำลังกิน
Eating7	นักปราชญ์คนที่ 7 กำลังกิน
Eating8	นักปราชญ์คนที่ 8 กำลังกิน
Eating9	นักปราชญ์คนที่ 9 กำลังกิน
Fork1	ส้อมคนที่ 1
Fork2	ส้อมคนที่ 2
Fork3	ส้อมคนที่ 3
Fork4	ส้อมคนที่ 4
Fork5	ส้อมคนที่ 5
Fork6	ส้อมคนที่ 6
Fork7	ส้อมคนที่ 7
Fork8	ส้อมคนที่ 8
Fork9	ส้อมคนที่ 9

เมื่อได้ออกแบบจำลองเพทรีเน็ตแล้ว ผู้วิจัยได้หาการเข้าถึงมาร์กกิงทั้งหมดของระบบ ซึ่งมีทั้งหมด 77 สถานะ จากนั้น ผู้วิจัยได้วาดกราฟปริภูมิสถานะของแบบจำลองเพทรีเน็ตออกมา โดยลักษณะของกราฟปริภูมิสถานะของระบบนี้มีความสลับซับซ้อนและการวนซ้ำเกิดขึ้น มีทั้งหมด 68 ชั้น อย่างไรก็ตาม ในรูปที่ 6.5 แสดงบางส่วนของกราฟปริภูมิสถานะเพื่อให้เห็นลักษณะของกราฟปริภูมิสถานะของระบบนี้ โดยการนำเสนอแผนภาพข้อมูล (Data visualization) ของกราฟปริภูมิสถานะของระบบนี้แสดงในภาคผนวก ค

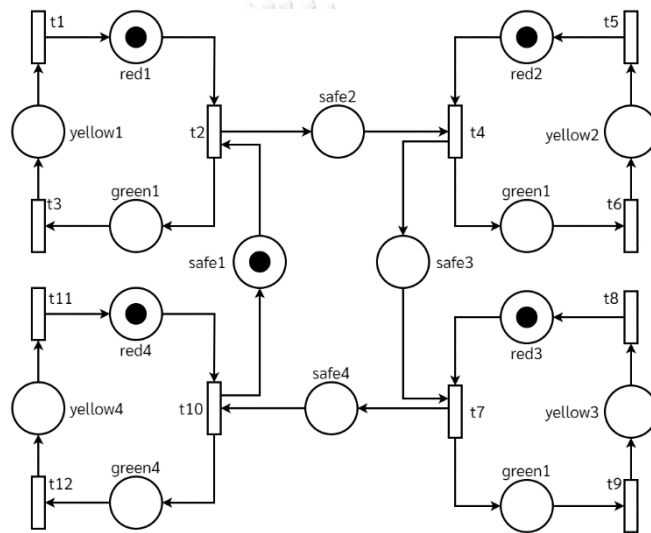


รูปที่ 6.5 ตัวอย่างกราฟปริภูมิสถานะของระบบการรับประทานอาหารของนักปราชญ์ 9 คน



### 6.1.3 กรณีศึกษาที่ 3 แบบจำลองเพทรีเน็ตสำหรับระบบสัญญาณไฟจราจรสี่ต้น

แบบจำลองเพทรีเน็ตสำหรับระบบสัญญาณไฟจราจรสี่ต้น แสดงในรูปที่ 6.6 ซึ่งไฟจราจรทุกต้นทำงานได้ก็ต่อเมื่ออยู่ในสถานะที่ปลอดภัย เริ่มจาก ไฟแดง ไฟเขียวและไฟเหลือง ตามลำดับ โดยเริ่มจากสัญญาณไฟจราจรต้นที่ 1 เปลี่ยนเป็นสีเขียวแล้ว ไฟจราจรต้นที่ 2 ถึงสามารถทำงานได้และเมื่อไฟจราจรต้นที่ 2 เปลี่ยนเป็นสีเขียว ไฟจราจรต้นที่ 3 ถึงสามารถทำงานได้และเมื่อไฟจราจรต้นที่ 3 เปลี่ยนเป็นสีเขียว ไฟจราจรต้นที่ 4 ถึงสามารถทำงานได้ เมื่อทุกเสาไฟจราจรเริ่มทำงาน ไฟจราจรก็สามารถเปลี่ยนเป็นสีเขียวหรือสีเหลืองได้โดยพร้อมกัน ซึ่งรายชื่อเพลสและคำอธิบายแสดงในตารางที่ 6.4



รูปที่ 6.6 แบบจำลองเพทรีเน็ตสำหรับระบบสัญญาณไฟจราจรสี่ต้น

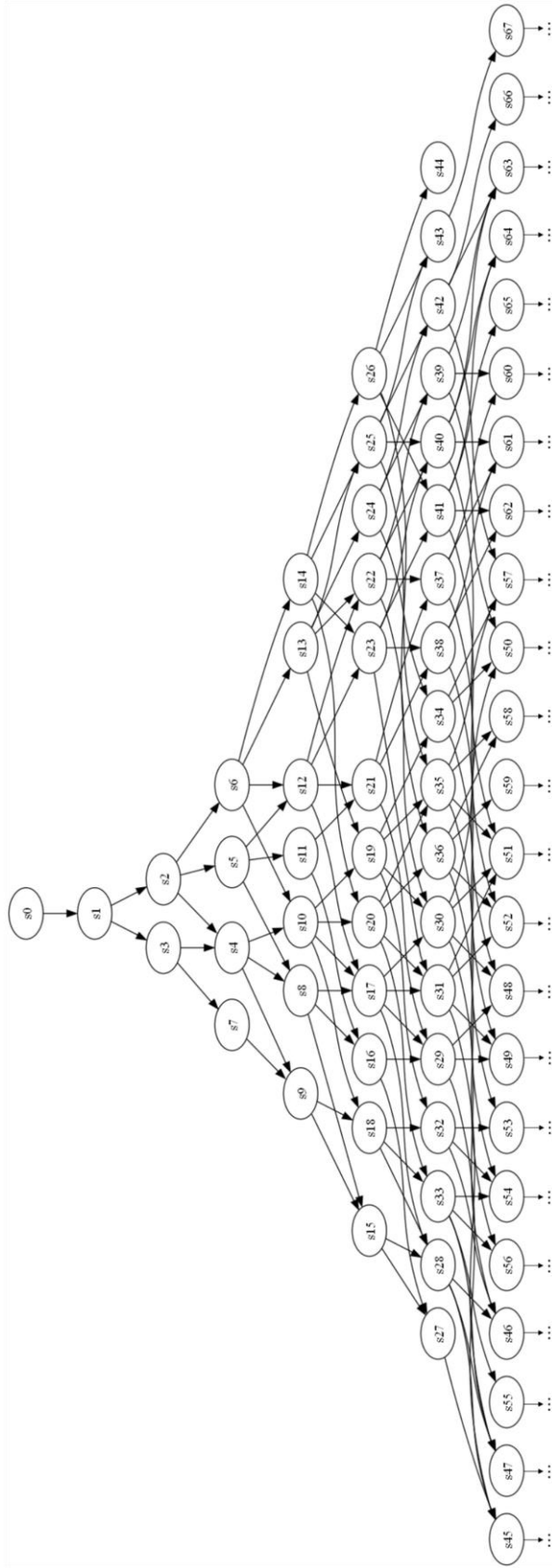
ตารางที่ 6.4 รายชื่อเพลสและคำอธิบายของกรณีศึกษาที่ 3 ระบบสัญญาณไฟจราจรสี่ต้น

ชื่อเพลส	คำอธิบาย
Red1	สัญญาณไฟแดงของเสาไฟจราจรต้นที่ 1
Red2	สัญญาณไฟแดงของเสาไฟจราจรต้นที่ 2
Red3	สัญญาณไฟแดงของเสาไฟจราจรต้นที่ 3
Red4	สัญญาณไฟแดงของเสาไฟจราจรต้นที่ 4
Green1	สัญญาณไฟเขียวของเสาไฟจราจรต้นที่ 1
Green2	สัญญาณไฟเขียวของเสาไฟจราจรต้นที่ 2
Green3	สัญญาณไฟเขียวของเสาไฟจราจรต้นที่ 3
Green4	สัญญาณไฟเขียวของเสาไฟจราจรต้นที่ 4

ตารางที่ 6.4 รายชื่อเพลสและคำอธิบายของกรณีศึกษาที่ 3 ระบบสัญญาณไฟจราจรสี่ต้น (ต่อ)

ชื่อเพลส	คำอธิบาย
Yellow1	สัญญาณไฟเหลืองของเสาไฟจราจรต้นที่ 1
Yellow2	สัญญาณไฟเหลืองของเสาไฟจราจรต้นที่ 2
Yellow3	สัญญาณไฟเหลืองของเสาไฟจราจรต้นที่ 3
Yellow4	สัญญาณไฟเหลืองของเสาไฟจราจรต้นที่ 4
Safe1	สถานะปลอดภัยของเสาไฟจราจรต้นที่ 1
Safe2	สถานะปลอดภัยของเสาไฟจราจรต้นที่ 2
Safe3	สถานะปลอดภัยของเสาไฟจราจรต้นที่ 3
Safe4	สถานะปลอดภัยของเสาไฟจราจรต้นที่ 4

เมื่อได้ออกแบบจำลองเพทรีเน็ตแล้ว ผู้วิจัยได้ทำการเข้าถึงมาร์กกิงทั้งหมดของระบบ ซึ่งมีทั้งหมด 325 สถานะ จากนั้น ผู้วิจัยได้วาดกราฟปริภูมิสถานะของแบบจำลองเพทรีเน็ตออกมา โดยลักษณะของกราฟปริภูมิสถานะของระบบนี้มีลักษณะขยายออกเป็นแนวกว้างและมีความซับซ้อน มีทั้งหมด 81 ชั้น อย่างไรก็ตาม ในรูปที่ 6.7 แสดงบางส่วนของกราฟปริภูมิสถานะเพื่อให้เห็นลักษณะของกราฟปริภูมิสถานะของระบบนี้ โดยการนำเสนอแผนภาพข้อมูลของกราฟปริภูมิสถานะของระบบนี้ แสดงในภาคผนวก ค

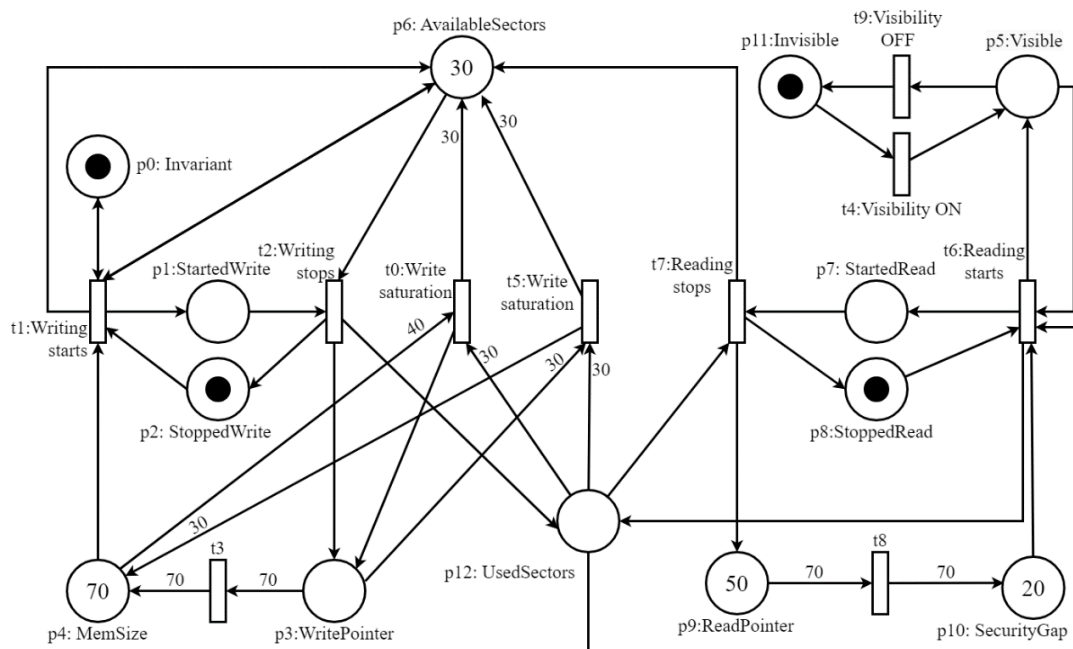


รูปที่ 6.7 ตัวอย่างกราฟประนัยสถานะของระบบสัญญาณไฟฟ้าจรัสสี่บิต



#### 6.1.4 กรณีศึกษาที่ 4 แบบจำลองเพทรีเน็ตสำหรับระบบหน่วยความจำดาวเทียม

แบบจำลองเพทรีเน็ตของระบบหน่วยความจำดาวเทียมจากการประกวด Model Checking Contest 2020 แสดงในรูปที่ 6.8 ที่รวบรวมพฤติกรรมการจัดการหน่วยความจำภายในดาวเทียม ที่มีการกำหนดขนาดของหน่วยความจำด้วยพารามิเตอร์  $X$  (ถูกจำลองโดยมาร์กกิงของเพลส  $p4$ ) ซึ่งใช้วิธีการจัดการข้อมูลแบบบัฟเฟอร์วงกลม (Circular buffer) ที่เข้าถึงได้ผ่านพอยน์เตอร์ 2 ตัว คือ พอยน์เตอร์การเขียน (Writing pointer หรือ WP) ซึ่งค่าหน่วยความจำจะถูกจำลองโดยมาร์กกิงของเพลส  $p3$  และพอยน์เตอร์การอ่าน (Reading pointer หรือ RP) ซึ่งค่าหน่วยความจำจะถูกจำลองโดยมาร์กกิงของเพลส  $p9$  โดยการเขียนข้อมูลลงในหน่วยความจำสามารถเขียนได้ตลอดเวลา ในขณะที่การอ่านข้อมูลจากหน่วยความจำสามารถอ่านได้ก็ต่อเมื่อมองเห็น (Visible) สถานีดาวนโหลด จากนั้นเพื่อป้องกันข้อผิดพลาดที่อาจเกิดขึ้นและคำนึงถึงการสูญเสียเซกเตอร์หน่วยความจำที่อาจเกิดขึ้นในระหว่างภารกิจ ผู้ควบคุมต้องตรวจสอบให้แน่ใจว่ามีช่องว่างด้านความปลอดภัยหรือบัฟเฟอร์ระหว่างพอยน์เตอร์การเขียนและพอยน์เตอร์การอ่านเสมอ กำหนดด้วยพารามิเตอร์  $Y$  (ถูกจำลองโดยมาร์กกิงของเพลส  $p10$ ) ซึ่งเพลส  $p0$  เป็นตัวยืนยง (Invariant) ที่ใช้ตรวจสอบช่องว่างด้านความปลอดภัยนี้เสมอ นอกจากนี้ เซกเตอร์ที่ใช้ได้ (Available sector) จะมีค่าเท่ากับ  $X-2*Y$  ซึ่งจะถูกจำลองโดยมาร์กกิงของเพลส  $p6$  ซึ่งรายชื่อเพลสและคำอธิบาย แสดงในตารางที่ 6.5

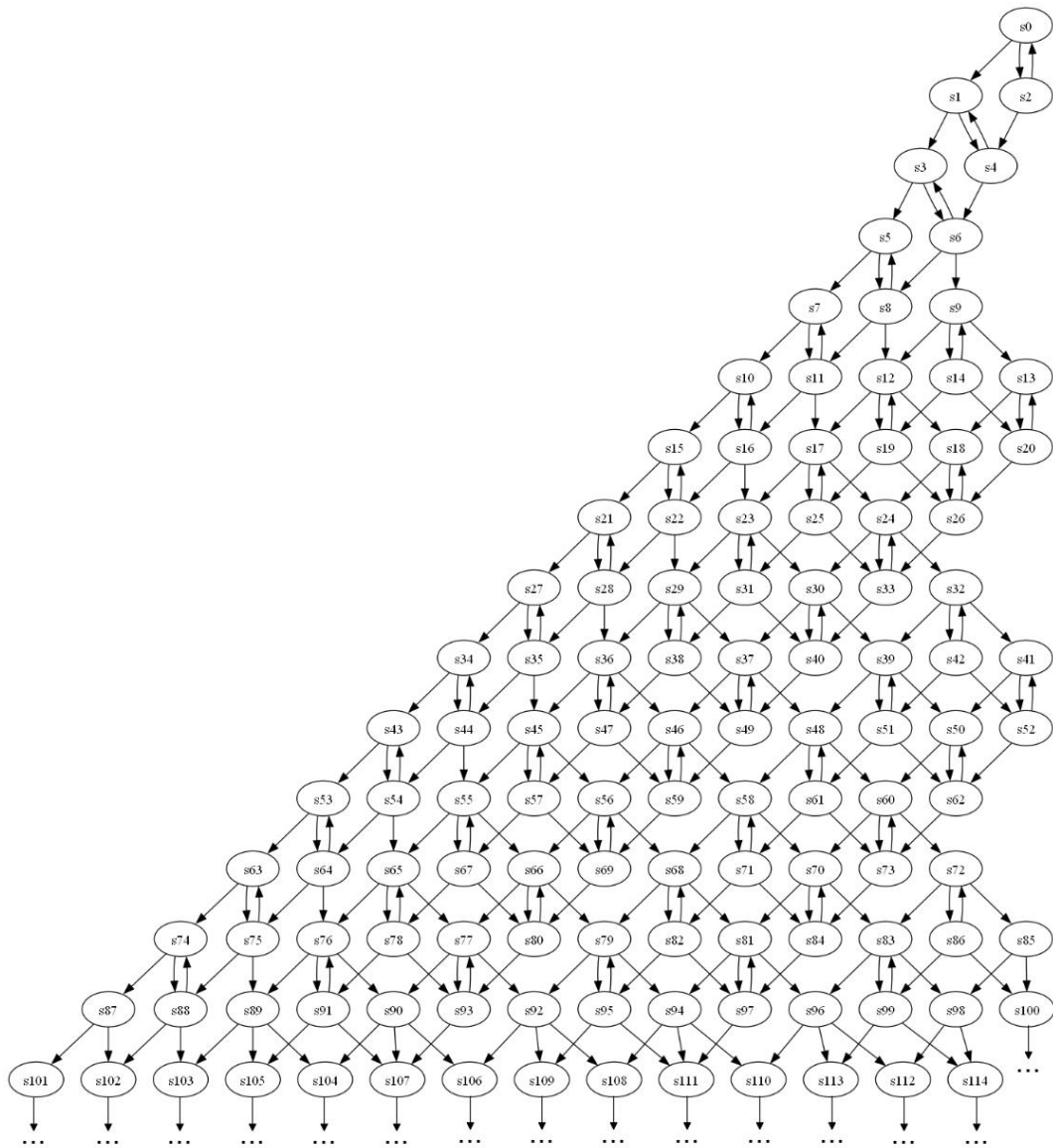


รูปที่ 6.8 แบบจำลองเพทรีเน็ตของระบบหน่วยความจำดาวเทียม

ตารางที่ 6.5 รายชื่อเฟลสและคำอธิบายของกรณีศึกษาที่ 4 ระบบหน่วยความจำดาวเทียม

ชื่อเฟลส	คำอธิบาย
p0: Invariant	ตัวยืนยันที่ใช้ตรวจสอบช่องว่างด้านความปลอดภัย
p1: StartedWrite	เริ่มการเขียนข้อมูลลงหน่วยความจำ
p2: StoppedWrite	หยุดการเขียนข้อมูลลงหน่วยความจำ
p3: WritePointer	พอยน์เตอร์การเขียน
p4: MemSize	หน่วยความจำที่เป็นตัวแทนของพารามิเตอร์ X
p5: Visible	สถานีดาวนโหลดเปิดการมองเห็น
p6: AvailableSectors	จำนวนเซกเตอร์ที่ใช้ได้
p7: StartedRead	เริ่มต้นการอ่านข้อมูลจากหน่วยความจำ
p8: StoppedRead	หยุดการอ่านข้อมูลจากหน่วยความจำ
p9: ReadPointer	พอยน์เตอร์การอ่าน
p10: SecurityGap	ขนาดช่องว่างด้านความปลอดภัยที่เป็นตัวแทนของพารามิเตอร์ Y
p11: Invisible	สถานีดาวนโหลดปิดการมองเห็น
p12: UsedSectors	จำนวนเซกเตอร์ที่ถูกใช้งาน

เมื่อได้ออกแบบจำลองเพทรีเน็ตแล้ว ผู้วิจัยได้ทำการเข้าถึงมาร์กกิงทั้งหมดของระบบ ซึ่งมีทั้งหมด 17,326 สถานะ จากนั้น ผู้วิจัยได้วาดกราฟปริภูมิสถานะของแบบจำลองเพทรีเน็ตออกมา โดยลักษณะของกราฟปริภูมิสถานะของระบบนี้มีลักษณะขยายออกเป็นแนวกว้างที่ลึกลงไปทั้งหมด 8,782 ชั้น อย่างไรก็ตาม ในรูปที่ 6.9 แสดงบางส่วนของกราฟปริภูมิสถานะ เพื่อให้เห็นลักษณะของกราฟปริภูมิสถานะของระบบนี้เท่านั้น โดยการนำเสนอแผนภาพข้อมูลของกราฟปริภูมิสถานะของระบบนี้แสดงในภาคผนวก ค



รูปที่ 6.9 ตัวอย่างกราฟปริภูมิสถานะของระบบหน่วยความจำดาวเทียม

## 6.2 ออกแบบวิธีการทดสอบ

ในขั้นตอนที่สอง ผู้วิจัยได้ออกแบบวิธีการทดสอบเป็น 3 ส่วน คือ ส่วนแรกเป็นการวัดความถูกต้องของเส้นทางดำเนินงานที่ถูกสร้างขึ้นโดยเครื่องสร้างปริภูมิสถานะ ส่วนที่สอง เป็นการทดสอบประสิทธิภาพการทำงานของเครื่องสร้างปริภูมิสถานะ โดยเปรียบเทียบกับแบบจำลองของแมชชีนเลิร์นนิ่งที่มีหรือแบบจำลองพื้นฐาน (Baseline model) ส่วนสุดท้ายเป็นการวัดประสิทธิภาพของเครื่องมือการทวนสอบคุณสมบัติการทำงานจากระบบ โดยการเปรียบเทียบประสิทธิภาพการทวนสอบกับผลการทวนสอบที่คาดหวัง

### 6.2.1 ประเมินความถูกต้องของการสร้างปฏิภูมิสถานะ

การประเมินแรก มีจุดประสงค์เพื่อวัดความถูกต้องของเครื่องสร้างปฏิภูมิสถานะ โดยการนำปฏิภูมิสถานะที่สร้างขึ้นมาเปรียบเทียบกับกราฟปฏิภูมิสถานะที่เป็นผลเฉลยด้วยการเดินไปบนกราฟ เริ่มจากสถานะเริ่มต้นจนถึงสถานะสุดท้ายจนครบทุกสถานะที่สามารถเข้าถึงได้ หากระบบที่นำมาทดสอบไม่มีสถานะสุดท้าย สถานะที่เกิดการวนซ้ำจะถือว่าเป็นสถานะสุดท้ายเช่นกัน

อย่างไรก็ตาม เมื่อนำเครื่องสร้างปฏิภูมิสถานะไปใช้งานจริง ปฏิภูมิสถานะที่ได้จะไม่ถูกต้องตรวจสอบอีกครั้ง เนื่องจาก ก่อนนำมาใช้งานจริง ผู้วิจัยได้ตรวจสอบความถูกต้องของปฏิภูมิสถานะทั้งหมดที่ถูกสร้างโดยเครื่องสร้างปฏิภูมิสถานะเรียบร้อยแล้ว

### 6.2.2 ประเมินประสิทธิภาพของเครื่องสร้างปฏิภูมิสถานะ

การประเมินส่วนที่สอง มีจุดประสงค์เพื่อวัดประสิทธิภาพของเครื่องสร้างปฏิภูมิสถานะที่สร้างมาจากแบบจำลอง LSTM-MLC และขีดแบ่งที่เหมาะสม โดยการประเมินประสิทธิภาพของเครื่องสร้างปฏิภูมิสถานะจากตัววัดการประเมินผลของการจำแนกหลายป้าย จากนั้น นำผลการประเมินมาเปรียบเทียบกับแบบจำลองของแมชชีนเลิร์นนิงที่มีหรือแบบจำลองพื้นฐานที่มีความสามารถในการจำแนกแบบหลายป้ายเพื่อเปรียบเทียบว่าเครื่องสร้างปฏิภูมิสถานะที่สร้างขึ้นมีประสิทธิภาพดีกว่าแบบจำลองพื้นฐานหรือไม่ โดยแบบจำลองพื้นฐานมี 3 แบบจำลอง ดังนี้

- 1) แบบจำลองความสัมพันธ์ไบนารี (Binary relevance model หรือ BR) [69] แบบจำลองนี้ใช้วิธีการแปลงการจำแนกแบบหลายป้ายให้เป็นป้ายกำกับป้ายเดียว (Single-label classification) ซึ่งแต่ละป้ายจะอิสระต่อกันและเป็นการจำแนกแบบไบนารี คือ 0 และ 1
- 2) แบบจำลองเพื่อนบ้านที่ใกล้ที่สุดแบบหลายป้าย (Multi-label K-nearest neighbor model หรือ ML-KNN) [70] แบบจำลองนี้ใช้อัลกอริทึมเพื่อนบ้านที่ใกล้ที่สุด (K-nearest neighbor algorithm) โดยการค้นหาเพื่อนบ้านที่ใกล้กันที่สุดสามารถมีได้มากกว่า 1 ป้ายและไม่เกินจำนวนค่า K ซึ่งเป็นจำนวนเพื่อนบ้านที่ใกล้กันที่สุด
- 3) แบบจำลองความสัมพันธ์ไบนารีแบบเพื่อนบ้านที่ใกล้ที่สุดแบบหลายป้าย (Binary relevance of K-nearest neighbors model หรือ BR-KNN) [71] แบบจำลองนี้เป็นการจำแนกแบบหลายป้ายคล้ายกับแบบจำลอง BR ที่นำอัลกอริทึมเพื่อนบ้านที่ใกล้ที่สุดมาใช้จำแนกแบบหลายป้าย

### 6.2.3 ประเมินประสิทธิภาพการทำงานของเครื่องมือการทวนสอบ

การประเมินส่วนที่สาม มีจุดประสงค์เพื่อวัดประสิทธิภาพของเครื่องมือการทวนสอบที่ทวนสอบคุณสมบัติการทำงานของระบบโดยเรียกใช้เครื่องสร้างปฏิภูมิสถานะ โดยผลการทวนสอบจาก

เครื่องมือการทวนสอบของงานวิจัยนี้เป็นผลลัพธ์ที่แท้จริง (Actual output) และผลการทวนสอบจากเครื่องมือที่น่าเป็นผลลัพธ์ที่คาดหวัง (Expected output)

สำหรับการประเมินประสิทธิภาพการทำงานของเครื่องมือการทวนสอบ ผู้วิจัยได้ออกแบบคุณสมบัติที่ต้องการทวนสอบโดยเขียนให้อยู่ในรูปแบบของสูตรแอลทีแอลที่ได้กำหนด ได้แก่ คุณสมบัติด้านความปลอดภัย คุณสมบัติด้านความดำเนินชีวิต คุณสมบัติด้านความสัมพันธ์และผู้วิจัยได้เพิ่มสูตรแอลทีแอลประเภท Until เพื่อให้ครอบคลุมตัวดำเนินงานเชิงเวลาทั้งหมด นอกจากนี้ ผู้วิจัยประเมินประสิทธิภาพในการตรวจสอบข้อมูลนำเข้าที่เป็นสูตรแอลทีแอลเพิ่มเติมอีกด้วย

### 6.3 ผลการดำเนินการทดสอบ

เมื่อได้ออกแบบกรณีศึกษาและออกแบบวิธีการทดสอบเรียบร้อยแล้ว ผู้วิจัยได้ดำเนินการทดลองสำหรับทุกกรณีศึกษาตามแบบการทดสอบ ได้ดังต่อไปนี้

#### 6.3.1 ผลทดสอบการประเมินความถูกต้องของการสร้างปริภูมิสถานะ

สำหรับการตรวจสอบความถูกต้องของการสร้างปริภูมิสถานะของระบบ ผู้วิจัยได้นำเครื่องสร้างปริภูมิสถานะมาสร้างกราฟปริภูมิสถานะที่อยู่ในรูปแบบของเส้นทางดำเนินงานย่อย ที่มีความยาวเท่ากับ 4 ซึ่งเมื่อนำเส้นทางย่อยมาต่อกันก็จะได้เป็นกราฟปริภูมิสถานะเช่นกัน จากนั้นนำเส้นทางย่อยที่ได้ไปเปรียบเทียบกับผลเฉลยที่เป็นกราฟปริภูมิสถานะต้นแบบด้วยวิธีการเดินไปบนกราฟต้นแบบ โดยเริ่มต้นจากสถานะเริ่มต้นไปจนครบทุกเส้นทางดำเนินงาน แสดงผลการประเมินได้ในตารางที่ 6.6 ซึ่งผลการประเมินจะแสดงจำนวนเส้นทางย่อยที่ทำนายไม่ถูกต้อง สามารถแบ่งออกเป็นผลการทำนายไม่ครบ หมายความว่า เครื่องสร้างปริภูมิสถานะสามารถทำนายสถานะออกมาได้ไม่ครบตามที่คาดหวังไว้ ซึ่งสถานะที่ถูกทำนายออกมาอาจทำได้ถูกต้องแต่ได้ไม่ครบตามที่ต้องการหรือสถานะที่ถูกทำนายออกมานั้นไม่ถูกต้องเลยและมีจำนวนน้อยกว่าที่คาดไว้ ในขณะที่ ผลการทำนายเกิน หมายความว่า มีสถานะที่ถูกทำนายออกมาผิดตำแหน่งของเส้นทางดำเนินงาน ซึ่งทำให้เกิดเส้นทางดำเนินงานที่ไม่ถูกต้อง ดังนั้น หากต้องการปรับปรุงประสิทธิภาพการสร้างปริภูมิสถานะ ทำได้โดยการฝึกอบรมเครื่องสร้างปริภูมิสถานะเพิ่มเติม

ตารางที่ 6.6 ผลการประเมินความถูกต้องของการสร้างปฏิภูมิสถานะโดยใช้เครื่องสร้างปฏิภูมิสถานะ

กรณีศึกษา	จำนวนเส้นทาง ย่อยทั้งหมด	จำนวนที่ทำนายไม่ถูกต้อง		
		ผลการ ทำนาย ไม่ครบ	ผลการ ทำนาย เกิน	รวมทั้งหมด
1. ระบบเครื่องรับจ่ายเงินอัตโนมัติ	113	0	0	0
2. ระบบการรับประทานอาหารของ นักปราชญ์ 9 คน	56,916	1	0	1
3. ระบบสัญญาณไฟจราจรสี่ด้าน	22,332	0	0	0
4. ระบบหน่วยความจำดาวเทียม	944,926	39,910	3,426	43,336

จากตารางที่ 6.6 ผลการประเมินความถูกต้องของการสร้างปฏิภูมิสถานะของกรณีศึกษาที่ 1 และ 3 สามารถสร้างปฏิภูมิสถานะได้ถูกต้อง ในขณะที่ กรณีศึกษาที่ 2 เมื่อตรวจสอบความถูกต้องของการสร้างปฏิภูมิสถานะแล้ว ได้ผลลัพธ์การทำนายไม่ครบตามที่คาดหวังไว้ 1 เส้นทางย่อย จากทั้งหมด 56,916 เส้นทางย่อย สำหรับกรณีศึกษาที่ 4 เมื่อตรวจสอบความถูกต้องของการสร้างปฏิภูมิสถานะแล้ว ได้ผลลัพธ์การทำนายไม่ครบตามที่คาดหวังไว้ 39,910 เส้นทางย่อย และได้ผลลัพธ์การทำนายเกินกว่าที่คาดหวังไว้ 3,426 เส้นทางย่อย จากทั้งหมด 944,926 เส้นทางย่อย

### 6.3.2 ผลการทดสอบการประเมินประสิทธิภาพของเครื่องสร้างปฏิภูมิสถานะ

สำหรับผลการประเมินประสิทธิภาพของเครื่องสร้างปฏิภูมิสถานะสำหรับแต่ละกรณีศึกษาถูกนำไปเปรียบเทียบกับแบบจำลองพื้นฐานมี 3 แบบจำลอง ได้แก่ แบบจำลอง BR แบบจำลอง ML-KNN และแบบจำลอง BR-KNN ซึ่งใช้ชุดข้อมูลการเรียนรู้ที่ได้จัดเตรียมไว้สำหรับแต่ละกรณีศึกษา และเป็นชุดข้อมูลเดียวกันกับแบบจำลอง LSTM-MLC ใช้ในการเรียนรู้ โดยนำไปเปรียบเทียบกับตัววัดการประเมินผลของการจำแนกหลายป้าย ได้แก่ ค่าระยะที่หายไป (Hamming loss หรือ HL) ค่าความเที่ยงระดับไมโคร (Micro-Precision หรือ MP) ค่าเรียกคืนระดับไมโคร (Micro-Recall หรือ MR) และเอฟวันระดับไมโคร (Micro-F1 หรือ MF1) โดยค่า HL ที่คาดหวังต้องมีค่าน้อยมากหรือมีค่าเข้าใกล้ 0 ในขณะที่ ค่า MP MR และ MF1 ที่คาดหวังต้องมีค่าเข้าใกล้ 1 หรือ 100%

จากผลการประเมินแสดงในตารางที่ 6.7 พบว่า เครื่องสร้างปฏิภูมิสถานะที่นำเสนอมีประสิทธิภาพดีกว่าแบบจำลองพื้นฐานทั้งหมด โดยรายละเอียดแบ่งตามกรณีศึกษา ดังต่อไปนี้

**กรณีศึกษาที่ 1** ระบบเครื่องรับจ่ายเงินอัตโนมัติ พบว่า เมื่อเปรียบเทียบกับแบบจำลอง BR เครื่องสร้างปฏิภูมิสถานะสามารถลด HL ลง 0.02117 และเพิ่ม MF1 ได้ถึง 0.32709 นอกจากนี้ เมื่อ

เปรียบเทียบกับแบบจำลอง ML-KNN ค่า HL ลดลง 0.01001 และ MF1 เพิ่มขึ้น 0.17271 เช่นเดียวกัน เมื่อเปรียบเทียบกับแบบจำลอง BR-KNN พบว่า ค่า HL ลดลงถึง 0.01674 และ MF1 เพิ่มขึ้นถึง 0.30540

**กรณีศึกษาที่ 2** ระบบการรับประทานอาหารของนักปราชญ์ 9 คน พบว่า เมื่อเปรียบเทียบกับแบบจำลอง BR เครื่องสร้างประมุขสถานะสามารถลด HL ลงไป 0.22872 และค่า MF1 สามารถเพิ่มขึ้นถึง 0.79861 จากนั้น เมื่อเปรียบเทียบกับแบบจำลอง ML-KNN พบว่าค่า HL ลดลงไป 0.08320 และ MF1 เพิ่มขึ้น 0.57868 ในทำนองเดียวกัน เมื่อเปรียบเทียบกับแบบจำลอง BR-KNN พบว่าสามารถลด HL ลงไปถึง 0.07130 และ MF1 ดีขึ้น 0.58621

**กรณีศึกษาที่ 3** ระบบสัญญาณไฟจราจรสี่ต้น พบว่า เมื่อเปรียบเทียบกับแบบจำลอง BR เครื่องสร้างประมุขสถานะสามารถลด HL ลงไป 0.01305 และค่า MF1 สามารถเพิ่มขึ้นถึง 0.93675 จากนั้น เมื่อเปรียบเทียบกับแบบจำลอง ML-KNN พบว่าค่า HL ลดลงไป 0.00972 และ MF1 เพิ่มขึ้น 0.92520 ในทำนองเดียวกัน เมื่อเปรียบเทียบกับแบบจำลอง BR-KNN พบว่าสามารถลด HL ลงไปถึง 0.00946 และ MF1 ดีขึ้น 0.96770

**กรณีศึกษาที่ 4** ระบบหน่วยความจำดาวเทียม พบว่า เมื่อเปรียบเทียบกับแบบจำลอง BR เครื่องสร้างประมุขสถานะสามารถลด HL ลงไปถึง 0.00026 และเป็นที่น่าประทับใจเมื่อค่า MF1 สามารถเพิ่มขึ้นได้มากถึง 0.96625 จากนั้น เมื่อเปรียบเทียบกับแบบจำลอง ML-KNN พบว่าค่า HL ลดลงไป 0.00249 และ MF1 เพิ่มขึ้น 0.95522 ในทำนองเดียวกัน เมื่อเปรียบเทียบกับแบบจำลอง BR-KNN พบว่าสามารถลด HL ลงไปถึง 0.00021 และ MF1 ดีขึ้นถึง 0.95693

ตารางที่ 6.7 ผลการประเมินประสิทธิภาพของเครื่องสร้างประมุขสถานะและแบบจำลองพื้นฐาน

กรณีศึกษา	แบบจำลอง	ตัววัดการประเมินผล			
		HL	MP	MR	MF1
1 ระบบเครื่องรับจ่ายเงินอัตโนมัติ (54 สถานะ)	BR	0.02190	0.68807	0.63830	0.66225
	ML-KNN	0.01074	0.95977	0.71064	0.81663
	BR-KNN	0.01747	0.87417	0.56170	0.68394
	LSTM-MLC	<b>0.00073</b>	<b>0.99145</b>	<b>0.98723</b>	<b>0.98934</b>
2 ระบบการรับประทานอาหารของนักปราชญ์ 9 คน (77 สถานะ)	BR	0.22873	0.18025	0.30779	0.20135
	ML-KNN	0.08321	0.43607	0.40747	0.42128
	BR-KNN	0.07131	0.49549	0.35735	0.41375
	LSTM-MLC	<b>0.00001</b>	<b>0.99998</b>	<b>0.99940</b>	<b>0.99996</b>

ตารางที่ 6.7 ผลการประเมินประสิทธิภาพของเครื่องสร้างปริภูมิสถานะและแบบจำลองพื้นฐาน (ต่อ)

กรณีศึกษา	แบบจำลอง	ตัววัดการประเมินผล			
		HL	MP	MR	MF1
3 ระบบสัญญาณไฟจราจรสี่ ต้น (325 สถานะ)	BR	0.01306	0.09563	0.04717	0.06318
	ML-KNN	0.00973	0.33371	0.04208	0.07473
	BR-KNN	0.00947	0.34688	0.0169	0.03223
	LSTM-MLC	<b>0.00001</b>	<b>0.99986</b>	<b>0.99999</b>	<b>0.99993</b>
4 ระบบหน่วยความจำ ดาวเทียม (17,326 states)	BR	0.00027	0.00281	0.00310	0.00274
	ML-KNN	0.00250	0.01978	0.01056	0.01377
	BR-KNN	0.00022	0.02118	0.00843	0.01206
	LSTM-MLC	<b>0.00001</b>	<b>0.98594</b>	<b>0.95416</b>	<b>0.96899</b>

นอกจากนั้น ผู้วิจัยได้นำแบบจำลอง BR ML-KNN และ BR-KNN มาประเมินความถูกต้องของการสร้างปริภูมิสถานะสำหรับกรณีศึกษาที่ 4 และนำมาเปรียบเทียบผลการประเมินความถูกต้องของแบบจำลอง LSTM-MLC แสดงได้ในตารางที่ 6.8

ตารางที่ 6.8 ผลการประเมินความถูกต้องของการสร้างปริภูมิสถานะโดยใช้เครื่องสร้างปริภูมิสถานะของแบบจำลองพื้นฐาน

กรณีศึกษา	แบบจำลอง	จำนวนที่ทำนายไม่ถูกต้อง		
		ผลการทำนายไม่ ครบ	ผลการ ทำนายเกิน	รวม ทั้งหมด
4 ระบบหน่วยความจำ ดาวเทียม (17,326 states) มีจำนวนเส้นทางย่อย ทั้งหมด <b>944,926</b> เส้นทาง	BR	-	-	-
	ML-KNN	321,529	24,358	<b>345,887</b>
	BR-KNN	338,997	7,173	<b>346,170</b>
	LSTM-MLC	39,910	3,426	<b>43,336</b>

จากตารางที่ 6.8 พบว่า แบบจำลองพื้นฐาน ML-KNN และ BR-KNN มีจำนวนเส้นทางดำเนินงานที่ทำนายไม่ถูกต้องเป็นจำนวนมากเมื่อเปรียบเทียบกับแบบจำลอง LSTM-MLC และในขณะที่แบบจำลอง BR ใช้เวลาในการตรวจสอบนาน ทำให้ไม่สามารถประเมินความถูกต้องได้ ดังนั้น แบบจำลอง LSTM-MLC สามารถทำนายเส้นทางดำเนินงานได้ถูกต้องกว่าแบบจำลองพื้นฐาน



### 6.3.3 ผลทดสอบการประสิทธิภาพการทำงานของเครื่องมือการทวนสอบ

การประเมินประสิทธิภาพการทำงานของเครื่องมือการทวนสอบเป็นการเปรียบเทียบผลการทวนสอบว่าเป็นไปตามผลลัพธ์ที่คาดหวังหรือไม่ โดยใช้คุณสมบัติแบบเฉพาะที่ได้นำเสนอก่อนหน้า คือ คุณสมบัติด้านความปลอดภัย คุณสมบัติด้านความดำเนินชีวิต คุณสมบัติด้านความสัมพันธ์และสูตรแอลทีแอลประเภท Until การเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง แบ่งตามกรณีศึกษา ดังต่อไปนี้

**กรณีศึกษาที่ 1** ระบบเครื่องรับจ่ายเงินอัตโนมัติ ผู้วิจัยได้ออกแบบสถานการณ์ที่ต้องการทวนสอบ และกำหนดสูตรแอลทีแอลใช้ชื่อของเพลสที่แสดงในตารางที่ 6.2 ได้ทั้งหมด 9 เทสเคส ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง แสดงในตารางที่ 6.9

ตารางที่ 6.9 ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง สำหรับกรณีศึกษาที่ 1

รหัส เทส เคส	สถานการณ์ที่ต้องการ ทวนสอบ	สูตรแอลทีแอล	ผลการทวน สอบที่ แท้จริง	ผลการ ทวนสอบที่ คาดหวัง	ผลการ เปรียบเทียบ
ATM- TC- 01	การถอนเงินสด การ ฝากเงินสด หรือจ่าย บิลไม่ทำงานพร้อมกัน	G-(p5 && p9 && p12)	Satisfied	Satisfied	ผ่าน
ATM- TC- 02	ในขณะที่เลือกการจ่าย บิล เงินสดจะไม่ถูก ถอนออกมา	G-(p12 && p8)	Satisfied	Satisfied	ผ่าน
ATM- TC- 03	PIN จะถูกตรวจสอบใน ที่สุด	F(p2)	Satisfied	Satisfied	ผ่าน
ATM- TC- 04	ในที่สุดแล้ว ผู้ใช้จะ ได้รับบัตร ATM คืน	F(p15)	Not Satisfied	Not Satisfied	ผ่าน
ATM- TC- 05	เมื่อใดก็ตามที่ผู้ใช้เลือก ฝากเงินสด จากนั้นเงิน สดจะถูกฝากในที่สุด	F(p9) => F((p9) && F(p11))	Satisfied	Satisfied	ผ่าน

ตารางที่ 6.9 ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง  
สำหรับกรณีศึกษาที่ 1 (ต่อ)

รหัส ทดสอบ เคส	สถานการณ์ที่ ต้องการทวนสอบ	สูตรแอลทีแอล	ผลการทวน สอบที่ แท้จริง	ผลการทวน สอบที่ คาดหวัง	ผลการ เปรียบเทียบ
ATM- TC-06	เมื่อใดก็ตามที่ใช้ เลือกจ่ายบิล จากนั้นการถอน เงินหรือการฝาก เงินจะเกิดขึ้นใน ที่สุด	$F(p_{12}) \Rightarrow F((p_{12}) \&\& F(p_5 \parallel p_9))$	Not Satisfied	Not Satisfied	ผ่าน
ATM- TC-07	เมื่อใดก็ตามที่ผู้ใช้ ใส่ PIN จากนั้น PIN ถูกตรวจสอบ ในเวลาถัดมา	$F(p_1) \Rightarrow F((p_1) \&\& X(p_2))$	Satisfied	Satisfied	ผ่าน
ATM- TC-08	เมื่อตรวจสอบ ยอดเงินคงเหลือ สำหรับการถอน แล้ว จากนั้นเงินสด จะถูกถอนออกมา ในเวลาถัดมา	$F(p_7) \Rightarrow F((p_7) \&\& X(p_8))$	Not Satisfied	Not Satisfied	ผ่าน
ATM- TC-09	สถานะของระบบ อยู่ในสถานะเริ่มต้น จนกระทั่งผู้ใช้ใส่ PIN	$p_0 \cup p_1$	Satisfied	Satisfied	ผ่าน

**กรณีศึกษาที่ 2** ระบบการรับประทานอาหารของนักปราชญ์ 9 คน ผู้วิจัยได้ออกแบบสถานการณ์ที่ต้องการทดสอบและกำหนดสูตรแอลทีแอลใช้ชื่อของเพลสที่แสดงในตารางที่ 6.3 ได้ทั้งหมด 10 เทส เคส ผลการเปรียบเทียบผลการทดสอบจากเครื่องมือการทดสอบกับผลลัพธ์ที่คาดหวัง แสดงในตารางที่ 6.10

ตารางที่ 6.10 ผลการเปรียบเทียบผลการทดสอบจากเครื่องมือการทดสอบกับผลลัพธ์ที่คาดหวัง สำหรับกรณีศึกษาที่ 2

รหัส เทส เคส	สถานการณ์ที่ ต้องการทดสอบ	สูตรแอลทีแอล	ผลการ ทวนสอบ ที่แท้จริง	ผลการ ทวนสอบ ที่คาดหวัง	ผลการ เปรียบเทียบ
DPS- TC- 01	เมื่อนักปราชญ์คนที่ 6 7 และ 8 ไม่สามารถกินพร้อมกันได้	G-(Eating6 && Eating7 && Eating8)	Satisfied	Satisfied	ผ่าน
DPS- TC- 02	เมื่อนักปราชญ์คนที่ 3 กำลังกิน ส้อมคนที่ 3 และ 4 จะต้องไม่ว่างเสมอ	G-(Eating3 && Fork3 && Fork4)	Satisfied	Satisfied	ผ่าน
DPS- TC- 03	ในที่สุดแล้วจะมีนักปราชญ์คนใดคนหนึ่งกำลังคิดอยู่	F(Thinking1    Thinking2    Thinking3    Thinking4    Thinking5    Thinking6    Thinking7    Thinking8    Thinking9)	Satisfied	Satisfied	ผ่าน
DPS- TC- 04	นักปราชญ์คนที่ 1 2 หรือ 3 จะได้กินในที่สุด	F(Eating1    Eating2    Eating3)	Not Satisfied	Not Satisfied	ผ่าน

ตารางที่ 6.10 ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง สำหรับกรณีศึกษาที่ 2 (ต่อ 1)

รหัส เทส เคส	สถานการณ์ที่ ต้องการทวนสอบ	สูตรแอลทีแอล	ผลการทวน สอบที่ แท้จริง	ผลการ ทวนสอบ ที่คาดหวัง	ผลการ เปรียบเทียบ
DPS- TC-05	ในที่สุดแล้วจะมี นักปราชญ์คนใดคน หนึ่งกำลังกินอยู่	F(Eating1    Eating2    Eating3    Eating4    Eating5    Eating6    Eating7    Eating8    Eating9)	Satisfied	Satisfied	ผ่าน
DPS- TC-06	เมื่อใดก็ตามที่ นักปราชญ์คนที่ 5 กินเสร็จ จากนั้น นักปราชญ์คนที่ 6 ก็จะกินในที่สุด	F(Eating5) => F((Eating5) && F(Eating6))	Not Satisfied	Not Satisfied	ผ่าน
DPS- TC-07	เมื่อใดก็ตามที่ นักปราชญ์คนที่ 9 กำลังกิน จากนั้น เขาก็จะสามารถกิน ต่อไปได้หรือจะ กลับไปคิดต่อได้	F(Eating9) => F((Eating9) && F(Eating9    Thinking9))	Satisfied	Satisfied	ผ่าน
DPS- TC-08	เมื่อใดก็ตามที่ นักปราชญ์คนที่ 2 กินเสร็จแล้ว จากนั้น ส้อมคนที่ 2 และ 3 จะถูกวาง และนักปราชญ์จะ กลับไปคิด	F(Eating2) => F((Eating2) && X(Thinking2 && Fork2 && Fork3))	Not Satisfied	Not Satisfied	ผ่าน

ตารางที่ 6.10 ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง สำหรับกรณีศึกษาที่ 2 (ต่อ 2)

รหัส เทส เคส	สถานการณ์ที่ ต้องการทวนสอบ	สูตรแอลทีแอล	ผลการทวน สอบที่ แท้จริง	ผลการ ทวนสอบ ที่คาดหวัง	ผลการ เปรียบเทียบ
DPS- TC-09	นักปราชญ์คนที่ 4 จะหยุดคิดเมื่อเขา กำลังกิน	Thinking4 U Eating4	Not Satisfied	Not Satisfied	ผ่าน
DPS- TC-10	นักปราชญ์คนที่ 4 จะหยุดกินเมื่อเขา กำลังคิด	Eating4 U Thinking4	Satisfied	Satisfied	ผ่าน

**กรณีศึกษาที่ 3** ระบบสัญญาณไฟจราจรสีต้น ผู้วิจัยได้ออกแบบสถานการณ์ที่ต้องการทวนสอบและกำหนดสูตรแอลทีแอลใช้ชื่อของเพลสที่แสดงในตารางที่ 6.4 ได้ทั้งหมด 9 เทสเคส ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง แสดงในตารางที่ 6.11

ตารางที่ 6.11 ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง สำหรับกรณีศึกษาที่ 3

รหัส เทส เคส	สถานการณ์ที่ต้องการ ทวนสอบ	สูตรแอลที แอล	ผลการทวน สอบที่ แท้จริง	ผลการทวน สอบที่ คาดหวัง	ผลการ เปรียบเทียบ
FTL- TC-01	สัญญาณไฟเขียวของ ทุกเสาไฟจราจรจะไม่ เกิดพร้อมกันเสมอ	G-(Green1 && Green2 && Green3 && Green4)	Not Satisfied	Not Satisfied	ผ่าน
FTL- TC-02	สัญญาณความ ปลอดภัยของแต่ละเสา ไฟจราจรจะไม่ทำงาน พร้อมกันเสมอ	G-(Safe1 && Safe2 && Safe3 && Safe4)	Satisfied	Satisfied	ผ่าน

ตารางที่ 6.11 ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง สำหรับกรณีศึกษาที่ 3 (ต่อ 1)

รหัส เทส เคส	สถานการณ์ที่ต้องการ ทวนสอบ	สูตรแอลที แอล	ผลการทวน สอบที่ แท้จริง	ผลการทวน สอบที่ คาดหวัง	ผลการ เปรียบเทียบ
FTL- TC-03	สัญญาณไฟจราจรต้นที่ 3 หรือต้นที่ 4 จะเป็น เปลี่ยนสีเขียวในที่สุด	F(Green3    Green4)	Satisfied	Satisfied	ผ่าน
FTL- TC-04	เมื่อสัญญาณไฟจราจร ต้นที่ 2 เป็นไฟเหลือง ไฟเขียวจะต้องไม่ติดใน ที่สุด	F(Yellow2 && -Green2)	Satisfied	Satisfied	ผ่าน
FTL- TC-05	เมื่อไรก็ตามที่เสาไฟ จราจรต้นที่ 1 เป็นสี แดงแล้ว จากนั้นไฟ จราจรจะเป็นสีแดงอีก ครั้งและปลอดภัยใน ที่สุด	F(Red1) => F((Red1) && F(Red1 && Safe1))	Satisfied	Satisfied	ผ่าน
FTL- TC-06	เมื่อไรก็ตามที่ เสาไฟ จ ร า จ ร ต้น ที่ 3 ปลอดภัยและไฟจราจร เป็นสีแดง จากนั้นไฟ เขียวจะทำงานใน ลำดับถัดมา	F(Safe3 && Red3) => F((Safe3 && Red3) && X(Green3))	Satisfied	Satisfied	ผ่าน
FTL- TC-07	เมื่อไรก็ตามที่ เสาไฟ จราจรต้นที่ 2 หรือ 3 ปลอดภัยแล้ว จากนั้น นั้นเสาไฟจราจรต้นที่ 4 ก็จะปลอดภัยในที่สุด	F(Safe2 && Safe3) => F((Safe2 && Safe3) && F(Safe4))	Satisfied	Satisfied	ผ่าน

ตารางที่ 6.11 ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง สำหรับกรณีศึกษาที่ 3 (ต่อ 2)

รหัส เทส เคส	สถานการณ์ที่ต้องการ ทวนสอบ	สูตรแอลที แอล	ผลการทวน สอบที่ แท้จริง	ผลการทวน สอบที่ คาดหวัง	ผลการ เปรียบเทียบ
FTL- TC-08	ไฟจรรยาต้นที่ 1 เป็นสี แดงและอยู่ในสถานะ ปลอดภัยจนกระทั่ง เปลี่ยนเป็นสีเขียว	(Red1 && Safe1) U Green1	Satisfied	Satisfied	ผ่าน
FTL- TC-09	ไฟจรรยาต้นที่ 4 เป็นสี แดง จนกระทั่งอยู่ใน สถานะปลอดภัยและ เปลี่ยนเป็นสีเขียว	Red4 U Green4	Satisfied	Satisfied	ผ่าน

**กรณีศึกษาที่ 4** ระบบหน่วยความจำดาวเทียม ผู้วิจัยได้ออกแบบสถานการณ์ที่ต้องการทวนสอบและกำหนดสูตรแอลทีแอลใช้ชื่อของเพลสที่แสดงในตารางที่ 6.5 ได้ทั้งหมด 13 เทสเคส ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง แสดงในตารางที่ 6.12

ตารางที่ 6.12 ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง สำหรับกรณีศึกษาที่ 4

รหัส เทส เคส	สถานการณ์ที่ต้องการ ทวนสอบ	สูตรแอลที แอล	ผลการทวน สอบที่ แท้จริง	ผลการ ทวนสอบที่ คาดหวัง	ผลการ เปรียบเทียบ
STM- TC-01	มีการตรวจสอบช่องว่าง ด้านความปลอดภัยเสมอ	G-(p0)	Satisfied	Satisfied	ผ่าน
STM- TC-02	ในที่สุดแล้วจำนวน เชกเตอร์ที่ใช้งานได้ต้องมี ค่าเท่ากับ X-2Y	F(p6*30)	Satisfied	Satisfied	ผ่าน

ตารางที่ 6.12 ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง สำหรับกรณีศึกษาที่ 4 (ต่อ 1)

รหัส เทส เคส	สถานการณ์ที่ต้องการ ทวนสอบ	สูตรแอลทีแอล	ผลการทวน สอบที่ แท้จริง	ผลการ ทวนสอบที่ คาดหวัง	ผลการ เปรียบเทียบ
STM- TC-03	ในที่สุดแล้วระบบจะ อ่าน ข้อมูล จาก หน่วยความจำได้	F(p7)	Not Satisfied	Not Satisfied	ผ่าน
STM- TC-04	เมื่อได้ก็ตามที่สถานี ดาวนโหลตเปิดการ มองเห็น จากนั้นระบบ จะอ่านข้อมูลในที่สุด	F(p5) => F((p5) && F(p7))	Not Satisfied	Not Satisfied	ผ่าน
STM- TC-05	เมื่อได้ก็ตามที่สถานี ดาวนโหลตเปิดการ มองเห็น จากนั้น การ อ่านจะหยุดหรือเริ่มอ่าน ข้อมูลได้ในเวลาถัด	F(p5) => F((p5) && X(p7    p8))	Satisfied	Satisfied	ผ่าน
STM- TC-06	เมื่อได้ก็ตามที่ระบบอ่าน หรือเขียนเสร็จแล้ว จากนั้น ระบบจะคืน พื้นที่ให้เซกเตอร์ที่ใช้ งานได้ให้มีค่าเท่ากับ 30 ในที่สุด	F(p1    p7) => F((p1    p7) && F(p6*30))	Satisfied	Satisfied	ผ่าน
STM- TC-07	เมื่อได้ก็ตามที่ระบบอ่าน ข้อมูลเสร็จแล้ว จากนั้น หน่วย ความ จำ คืน หน่วยความจำให้มีค่า เท่ากับ 69 ได้ในที่สุด	F(p1) => F((p1) && F(p4*69))	Satisfied	Satisfied	ผ่าน



ตารางที่ 6.12 ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง สำหรับกรณีศึกษาที่ 4 (ต่อ 2)

รหัส เทส เคส	สถานการณ์ที่ต้องการ ทวนสอบ	สูตรแอลที่ แอล	ผลการทวน สอบที่ แท้จริง	ผลการทวน สอบที่ คาดหวัง	ผลการ เปรียบเทียบ
STM- TC-08	เมื่อใดก็ตามที่หน่วยความจำมีขนาดเท่ากับ 70 และขนาดช่องว่างความปลอดภัยเท่ากับ 20 แล้ว จากนั้นจำนวนเซกเตอร์ที่ใช้งานได้จะมีค่าเท่ากับ 30 ในเวลาถัดมา	$F(p4*70 \&\& p10*20) \Rightarrow F((p4*70 \&\& p10*20) \&\& X(p6*30))$	Satisfied	Satisfied	ผ่าน
STM- TC-09	เมื่อใดก็ตามที่พอยน์เตอร์การเขียนเท่ากับ 70 แล้ว จากนั้น ขนาดของช่องว่างความปลอดภัยต้องมีค่าเท่ากับ 70	$F(p9*70) \Rightarrow F((p9*70) \&\& F(p10*70))$	Not Satisfied	Not Satisfied	ผ่าน
STM- TC-10	เมื่อใดก็ตามที่พอยน์เตอร์การเขียนเท่ากับ 70 แล้ว ขนาดของช่องว่างความปลอดภัยต้องมีค่าไม่เท่ากับ 70 และการอ่านยังหยุดอยู่	$F(p9*70) \Rightarrow F((p9*70) \&\& F(-p10*70 \&\& p8))$	Satisfied	Satisfied	ผ่าน
STM- TC-11	มีการตรวจสอบช่องว่างด้านความปลอดภัยเสมอไม่ว่าจะอ่านข้อมูล เขียนข้อมูลหรือสถานะการดาวน์โหลดปิดอยู่	$p0 \cup (p1 \parallel p7 \parallel p11)$	Satisfied	Satisfied	ผ่าน

ตารางที่ 6.12 ผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบกับผลลัพธ์ที่คาดหวัง สำหรับกรณีศึกษาที่ 4 (ต่อ 3)

รหัส เทศ เคส	สถานการณ์ที่ต้องการ ทวนสอบ	สูตรแอลที แอล	ผลการทวน สอบที่ แท้จริง	ผลการทวน สอบที่ คาดหวัง	ผลการ เปรียบเทียบ
STM- TC-12	ระบบจะหยุดอ่านข้อมูล จนกระทั่งสถานีดาวน โหลดเปิดการมองเห็นหรือ เริ่มต้นอ่านข้อมูลจาก หน่วยความจำ	p8 U (p5    p7)	Not Satisfied	Not Satisfied	ผ่าน
STM- TC-13	ระบบจะหยุดอ่านข้อมูล จนกระทั่งระบบเขียน ข้อมูลลงหน่วยความจำ หรือสถานีดาวนโหลดเปิด การมองเห็นหรืออ่าน ข้อมูลจากหน่วยความจำ	p8 U (p1    p5    p7)	Satisfied	Satisfied	ผ่าน

จากผลการเปรียบเทียบผลการทวนสอบจากเครื่องมือการทวนสอบและผลลัพธ์ที่คาดหวังที่ได้จากเครื่องมือที่นำ ที่แสดงในตารางที่ 6.9 ตารางที่ 6.10 ตารางที่ 6.11 และตารางที่ 6.12 พบว่าผลการทวนสอบที่แท้จริง ซึ่งเป็นผลการทวนสอบจากเครื่องมือการทวนสอบของงานวิจัยนี้ มีผลลัพธ์ที่ตรงกับผลการทวนสอบที่คาดหวัง ซึ่งเป็นผลการทวนสอบจากเครื่องมือที่นำ ซึ่งแต่ละเทศเคสเป็นสถานการณ์ที่ถูกจำลองขึ้นมาเพื่อใช้ในการประเมินประสิทธิภาพการทำงานของเครื่องมือการทวนสอบเท่านั้น เช่น กรณีศึกษาที่ 3 ระบบสัญญาณไฟจราจร 4 ต้น ผู้วิจัยจำลองสถานการณ์ที่ระบบทำงานผิดพลาด โดยให้ไฟเขียวเกิดขึ้นพร้อมกันเพื่อเป็นการทดสอบกรณีของการทวนสอบคุณสมบัติด้านความปลอดภัย เป็นต้น

นอกจากนี้ การประเมินประสิทธิภาพการตรวจสอบความถูกต้องของข้อมูลนำเข้าโดยใช้เครื่องมือการทวนสอบแบ่งออกตามคุณสมบัติที่กำหนด คือ ความปลอดภัย ความดำเนินชีวิตและความสัมพันธ์ รวมถึงสูตรแอลทีแอลประเภท Until โดยใช้กรณีศึกษาที่ 1 ใดๆก็ดี สำหรับกรณีศึกษาอื่นๆ ไม่จำเป็นต้องทำการทดลองสำหรับการตรวจสอบข้อมูลนำเข้า เนื่องจากวิธีการตรวจสอบข้อมูลนำเข้าเป็นวิธีการเดียวกัน จากการทดสอบ พบว่า เครื่องมือการทวนสอบสามารถ

ตรวจสอบความถูกต้องของข้อมูลนำเข้าได้อย่างถูกต้องตามที่คาดหวัง แสดงผลการประเมินใน  
ภาคผนวก ง



## บทที่ 7

### สรุปผลการวิจัยและข้อเสนอแนะ

#### 7.1 สรุปผลการวิจัย

งานวิจัยนี้เป็นการศึกษาเกี่ยวกับการผสมผสานแนวคิดของแอลเอสทีเอ็มและการจำแนกแบบหลายป้ายเพื่อพัฒนาเป็นเครื่องสร้างปริภูมิสถานะ (State space generator) สำหรับการทำโมเดลเชิงกึ่งแบบแอลทีแอล เพื่อบรรเทาการเกิดการระเบิดของปริภูมิสถานะและได้นำเสนอวิธีการหาขีดแบ่งที่เหมาะสมที่ช่วยให้แบบจำลองแอลเอสทีเอ็มมีความสามารถในการจำแนกแบบหลายป้ายได้แม่นยำมากยิ่งขึ้น ซึ่งเครื่องสร้างปริภูมิสถานะสามารถรับข้อมูลนำเข้าเป็นเส้นทางดำเนินงานที่มีความยาวไม่เกินขนาดของหน้าต่าง  $K-1$  สร้างปริภูมิสถานะได้ตามที่ต้องการ (On-demand) ซึ่งในงานวิจัยนี้ค่า  $K-1$  จะมีค่าเท่ากับ 4 เพื่อทำนายหาสถานะถัดไปได้ โดยไม่จำเป็นที่จะต้องเริ่มที่สถานะเริ่มต้นทุกครั้ง

นอกจากนี้ ในงานวิจัยนี้ได้ศึกษาเกี่ยวกับการทวนสอบคุณสมบัติการทำงานของระบบซอฟต์แวร์ เพื่อออกแบบและพัฒนาเครื่องมือการทวนสอบคุณสมบัติการทำงานของระบบ ได้แก่ คุณสมบัติด้านความปลอดภัย (Safety property) คุณสมบัติด้านความดำเนินชีวิต (Liveness property) คุณสมบัติด้านความสัมพันธ์ (Correlation property) และสูตรแอลทีแอลประเภท Until โดยใช้เทคนิคแบบออน-เดอะ-ฟลาย (On-the-fly) ที่มีการเรียกใช้เครื่องสร้างปริภูมิสถานะที่ถูกพัฒนามาก่อนหน้า

ดังนั้น เครื่องสร้างปริภูมิสถานะและเครื่องมือการทวนสอบคุณสมบัติการทำงานของระบบได้ถูกพัฒนาเป็นระบบที่มีความสามารถในการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติการทำงานของระบบ พร้อมทั้งรายงานผลว่าระบบที่ทวนสอบมีคุณสมบัติที่คาดหวังหรือไม่

การประเมินประสิทธิภาพการทำงานของเครื่องสร้างปริภูมิสถานะและการทวนสอบโดยใช้เครื่องสร้างปริภูมิสถานะใช้วิธีการทดสอบกับแบบจำลองเพทรีเน็ตที่จำลองการทำงานของระบบที่แตกต่างกัน 4 กรณีศึกษา เมื่อคุณลักษณะของกราฟปริภูมิสถานะของแต่ละกรณีศึกษา สามารถจำแนกคุณลักษณะตามโทโพโลยีของกราฟ (Topology of graph) ได้คือ

- 1) ระบบเครื่องรับจ่ายเงินอัตโนมัติ มีลักษณะกราฟปริภูมิสถานะแบบที่มีสถานะสุดท้าย มีการวนซ้ำแต่มีขนาดเล็กและไม่ซับซ้อน ความลึกเพียง 12 ชั้น
- 2) ระบบการรับประทานอาหารของนักปราชญ์ 9 คน มีลักษณะกราฟปริภูมิสถานะแบบไม่มีสถานะสุดท้าย มีการวนซ้ำและซับซ้อนลึกลงไปถึง 68 ชั้น
- 3) สัญญาณไฟจราจรสี่ด้าน มีลักษณะกราฟปริภูมิสถานะแบบไม่มีสถานะสุดท้าย มีการวนซ้ำและซับซ้อนลึกลงไปถึง 81 ชั้น โดยลักษณะของกราฟจะขยายออกเป็นแนวกว้าง

4) ระบบหน่วยความจำดาวเทียม ลักษณะกราฟปริภูมิสถานะแบบไม่มีสถานะสุดท้าย มีการวนซ้ำและซับซ้อนลึกลงไปถึง 8,782 ชั้น โดยกราฟมีขนาดใหญ่เกินกว่าที่จะสามารถวาดออกมาได้ ซึ่งกรณีนี้เป็นกรณีที่ใช้สำหรับทดสอบสำหรับจำนวนสถานะที่มากกว่า 10,000 สถานะ

สำหรับการทดสอบเน้นไปที่การประเมินประสิทธิภาพการทำงานและความถูกต้องสำหรับทุกกรณีศึกษา โดยมีรายละเอียดดังต่อไปนี้

1) จากผลการประเมินความถูกต้องของการสร้างปริภูมิสถานะโดยเครื่องสร้างปริภูมิสถานะพบว่า เครื่องสร้างปริภูมิสถานะสำหรับกรณีศึกษาที่ 1 และ 3 สามารถสร้างปริภูมิสถานะได้ถูกต้อง ในขณะที่ กรณีศึกษาที่ 2 จากเส้นทางย่อยทั้งหมด 56,916 เส้นทางย่อย เมื่อดำเนินการทดสอบพบว่า มี 1 เส้นทางย่อยที่ไม่สามารถทำนายสถานะได้ครบตามที่คาดหวัง และในกรณีศึกษาที่ 4 จากเส้นทางย่อยทั้งหมด 944,926 เส้นทางย่อย เมื่อดำเนินการทดสอบ พบว่ามี 3,426 เส้นทางย่อย ที่ไม่สามารถทำนายสถานะครบตามที่คาดหวังและมี 39,910 เส้นทางย่อย ที่ทำนายสถานะเกินกว่าที่คาดหวัง

2) จากผลการประเมินประสิทธิภาพของเครื่องสร้างปริภูมิสถานะจากตัววัดการประเมินผลของการจำแนกหลายป้ายที่ใช้การเปรียบเทียบกับแบบจำลองพื้นฐานที่ความสามารถจัดการจำแนกแบบหลายป้าย พบว่าเครื่องสร้างปริภูมิสถานะมีประสิทธิภาพดีกว่าแบบจำลองพื้นฐานอื่นๆ

3) จากผลการประเมินประสิทธิภาพของเครื่องมือการทวนสอบคุณสมบัติการทำงานของระบบที่นำเสนอ พบว่า ผลลัพธ์การทวนสอบทดสอบทั้งหมดของทุกกรณีศึกษาโดยใช้เครื่องมือการทวนสอบที่นำเสนอมีผลลัพธ์การทวนสอบที่ตรงกับผลลัพธ์การทวนสอบที่คาดหวัง

สรุปได้ว่า เครื่องสร้างปริภูมิสถานะที่นำเสนอนี้พัฒนาจากวิธีการแบบใหม่โดยใช้แนวคิดของแอลเอสทีเอ็มและการจำแนกแบบหลายเพื่อบรรเทาปัญหาการระเบิดของปริภูมิสถานะ มีความสามารถในการสร้างปริภูมิสถานะได้ตามต้องการและถูกนำไปใช้ในการทวนสอบคุณสมบัติการทำงานของระบบที่ใช้การกำหนดสูตรแอลเอสทีเอ็มเฉพาะโดยใช้เทคนิคแบบออน-เดอะ-ฟลายได้อย่างไรก็ตาม จากการประเมินความถูกต้องของการสร้างปริภูมิสถานะพบว่า มีบางกรณีศึกษาที่มีความผิดพลาดอยู่ ดังนั้น หากต้องการปรับปรุงความถูกต้องสามารถทำได้โดยการนำแบบเครื่องสร้างปริภูมิสถานะไปทำการฝึกอบรมให้มากยิ่งขึ้น

### 7.1.1 อภิปรายผลเพิ่มเติม

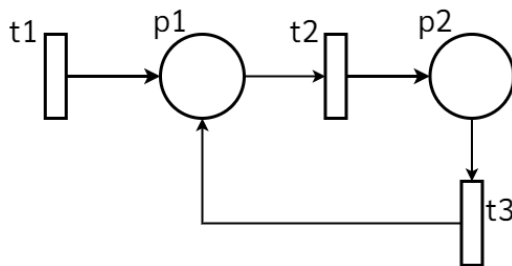
ในส่วนนี้เป็นการอภิปรายเพิ่มเติมเกี่ยวกับการนำเครื่องมือการทวนสอบคุณสมบัติที่งานวิจัยนี้ นำเสนอมาเปรียบเทียบกับโมเดลเชิงกึ่งที่มีอยู่ ได้แก่ เครื่องมือที่นำ และการเปรียบเทียบผลการ

ประเมินความถูกต้องของการสร้างปริภูมิสถานะสำหรับการฝึกศึกษาที่ 4 โดยมีการอธิบายเพิ่มเติมดังต่อไปนี้

1) เครื่องมือที่น่ามีโอกาสที่จะเกิดการระเบิดของปริภูมิสถานะหรือไม่

จากการทดลอง พบว่า เครื่องมือที่น่าไม่สามารถรองรับกรณีที่เกิดการสร้างโทเคนแบบอัตโนมัติ (Auto generate token) คือ การสร้างโทเคนออกมาเรื่อย ๆ โดยเริ่มจากทรานซ์ซัน แสดงได้ในรูปที่ 7.1 เห็นได้ว่า เมื่อระบบเริ่มทำงานทรานซ์ซัน t1 จะยิงโทเคนไปยังเพลส p1 ตลอดเวลา ทำให้เพลส p1 มีจำนวนโทเคนเพิ่มขึ้นเรื่อย ๆ ซึ่งเมื่อนำกรณีนี้จำลองบนเครื่องมือที่น่าและเรียกการทวนสอบ ผลปรากฏว่า เครื่องมือที่น่าแจ้งเตือนข้อผิดพลาดและทำให้ไม่สามารถใช้งานในกรณีนี้ได้

อย่างไรก็ดี เครื่องมือการทวนสอบของงานวิจัยนี้มีการกำหนดความลึกของการทวนสอบได้ตามต้องการทำให้สามารถทวนสอบกรณีนี้ได้



รูปที่ 7.1 แบบจำลองเพทรีเน็ตสำหรับกรณีของการสร้างโทเคนแบบอัตโนมัติ

2) ทำไมควรเลือกใช้เครื่องมือการทวนสอบของงานวิจัยนี้แทนเครื่องมือที่น่า

จากผลการทดสอบประสิทธิภาพการทำงาน of เครื่องมือการทวนสอบคุณสมบัติของงานวิจัยนี้ พบว่า เครื่องมือการทวนสามารถทวนสอบสูตรแอลทีแอลทีที่กำหนดได้ตรงตามผลลัพธ์ที่คาดหวัง ซึ่งเป็นผลการทวนสอบสูตรแอลทีแอลบนเครื่องมือที่น่า ซึ่งเมื่อลองสังเกตกรณีที่ไม่เป็นที่พอใจแล้วพบว่า เครื่องมือที่น่าไม่ได้แสดงตัวอย่างค่านที่ชัดเจนว่า เส้นทางที่นำไปสู่สถานะที่เกิดข้อผิดพลาดอยู่ในเส้นทางใดบ้าง ทำให้ผู้ใช้ต้องใช้วิธีการเดินไปบนกราฟปริภูมิสถานะด้วยมือ โดยเริ่มตั้งแต่สถานะเริ่มต้นจนถึงสถานะที่เกิดข้อผิดพลาดด้วยตัวเอง

ในขณะที่ เครื่องมือการทวนสอบของงานวิจัยนี้แสดงตัวอย่างค่านที่แสดงถึงเส้นทางดำเนินงานที่เกิดข้อผิดพลาดขึ้น จากนั้น ผู้ใช้สามารถใช้ความสามารถของเครื่องสร้างปริภูมิสถานะมาสร้างเส้นทางดำเนินงาน ณ ตำแหน่งของสถานะที่ผู้ใช้ต้องการโดยไม่จำเป็นต้องเริ่มจากสถานะเริ่มต้นเพื่อนำเส้นทางเหล่านั้นมาทำการวิเคราะห์และปรับปรุงระบบมีความถูกต้องมากยิ่งขึ้น

3) นำแบบจำลองพื้นฐานมาประเมินความถูกต้องของการสร้างปริภูมิสถานะสำหรับกรณีที่ 4 และนำมาเปรียบเทียบกับผลการประเมินความถูกต้องกับแบบจำลอง LSTM-MLC

จากกรณีศึกษาที่ 4 เป็นกรณีศึกษาที่อยู่ในขอบเขตที่ตั้งไว้ คือ มีจำนวนสถานะมากกว่า 5,000 สถานะ โดยกรณีศึกษานี้มีจำนวนสถานะทั้งหมด 17,326 สถานะ เมื่อนำข้อมูลที่ใช้ในการประเมินความถูกต้องแบบเดียวกับแบบจำลอง LSTM-MLC ในแบบจำลองพื้นฐานทั้ง 3 แบบจำลอง ได้แก่ แบบจำลอง BR แบบจำลอง ML-KNN และแบบจำลอง BR-KNN ที่แสดงผลการประเมินในตารางที่ 6.8 พบกว่า แบบจำลอง ML-KNN และ BR-KNN มีจำนวนเส้นทางดำเนินงานน้อยที่ผิดพลาดปริมาณมากและสำหรับแบบจำลอง BR ใช้เวลาในการตรวจสอบความถูกต้องนานทำให้ไม่สามารถประเมินความถูกต้องออกมาได้ เมื่อนำมาเปรียบเทียบกับแบบจำลอง LSTM-MLC พบกว่าแบบจำลอง LSTM-MLC มีเส้นทางดำเนินงานที่ทำนายได้ผิดพลาดเช่นเดียวกันแต่มีปริมาณที่น้อยกว่าอย่างเห็นได้ชัด

สรุปได้ว่า แบบจำลอง LSTM-MLC สามารถทำนายเส้นทางดำเนินงานได้ถูกต้องดีกว่าแบบจำลองพื้นฐานอื่นสำหรับการประเมินความถูกต้องของการสร้างปริภูมิสถานะสำหรับกรณีศึกษาที่ 4

## 7.2 ข้อจำกัดของงานวิจัย

แม้ว่าเครื่องสร้างปริภูมิสถานะนำเสนอมีประสิทธิภาพในการเรียนรู้ลำดับของเส้นทางดำเนินงานในปริภูมิสถานะได้ดีกว่าแบบจำลองพื้นฐานอื่น อย่างไรก็ตาม การสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติการทำงานของระบบโดยใช้เครื่องสร้างปริภูมิสถานะยังมีข้อจำกัด ดังต่อไปนี้

1) โดยลักษณะทั่วไปของแมชชีนเลิร์นนิงไม่สามารถรับประกันความแม่นยำได้ 100% สำหรับความแม่นยำในการทำนายผล ซึ่งต่างกับลักษณะทั่วไปของโมเดลเชิงกึ่งแบบดั้งเดิม ด้วยเหตุนี้จึงไม่แนะนำให้ใช้กับระบบที่สำคัญ (Critical systems)

2) ในกระบวนการทวนสอบ เครื่องมือการทวนสอบที่เรียกใช้เครื่องสร้างปริภูมิสถานะสามารถทำการทวนสอบสามารถทวนสอบคุณสมบัติได้ทันที โดยไม่จำเป็นต้องสร้างปริภูมิสถานะใหม่ทุกครั้ง ซึ่งแตกต่างกับโมเดลเชิงกึ่งแบบดั้งเดิม คือ ปริภูมิสถานะถูกสร้างขึ้นทุกครั้งเมื่อทำการทวนสอบระบบ ซึ่งอาจทำให้เสียเวลาในทุกครั้งที่ทำการทวนสอบ อย่างไรก็ตาม หากมีการแก้ไขการทำงานของระบบที่ต้องการทวนสอบ เครื่องสร้างปริภูมิสถานะต้องได้รับการฝึกอบรมอีกครั้ง

3) ในงานวิจัยนี้ได้เสนอวิธีการบรรเทาปัญหาการระเบิดของปริภูมิสถานะสำหรับระบบที่มีความซับซ้อนหรือมีขนาดใหญ่ โดยนำปริภูมิสถานะแบบนามธรรมมาแปลงให้อยู่ในรูปแบบของกราฟปริภูมิสถานะ จากนั้นใช้วิธีการเลื่อนไปบนหน้าตาตามขนาดของหน้าตา คือ ค่า  $K-1$  ที่เล็กที่สุดที่ทำให้แบบจำลอง LSTM-MLC สามารถเรียนรู้ได้อย่างมีประสิทธิภาพ ถ้าได้แบบจำลอง LSTM-MLC ที่มีประสิทธิภาพในการเรียนรู้ ก็จะทำให้ได้เป็นเครื่องสร้างปริภูมิสถานะ อย่างไรก็ตาม หากค่า  $K-1$  ไม่สอดคล้องกับชุดข้อมูล ก็อาจเป็นอุปสรรคต่อประสิทธิภาพการเรียนรู้ของแบบจำลอง LSTM-MLC เช่นกัน

4) ในงานวิจัยนี้ได้กำหนดสูตรแอลทีแอลแบบเฉพาะสำหรับคุณสมบัติที่กำหนด คือ คุณสมบัติด้านความปลอดภัย คุณสมบัติด้านความดำเนินชีวิต คุณสมบัติด้านความสัมพันธ์และสูตรแอลทีแอลประเภท Until ซึ่งสูตรแอลทีแอลที่กำหนดแบบเฉพาะยังไม่มีควมยืดหยุ่นมากนักและสูตรแอลทีแอลกำหนดโดยใช้ชื่อของเพลสเท่านั้น

### 7.3 ข้อเสนอแนะและแนวทางในการพัฒนา

การผสมผสานแนวคิดของแอลเอสทีเอ็มและการจำแนกแบบหลายป้ายสำหรับพัฒนาเป็นเครื่องสร้างปริภูมิสถานะ (State space generator) สำหรับการทำให้โมเดลเชิงกึ่งแบบแอลทีแอลที่นำเสนออาจไม่รองรับการทำงานของบางระบบหรือมีข้อจำกัดบางอย่าง ดังนั้น เพื่อเพิ่มประสิทธิภาพการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติโดยการกำหนดสูตรแอลทีแอล อาจต้องพิจารณาปัจจัยอื่นๆ เพิ่มเติม ได้แก่

1) ปรับปรุงอัลกอริทึมการหาขีดแบ่งที่เหมาะสมเพื่อเพิ่มประสิทธิภาพความแม่นยำสำหรับสร้างปริภูมิสถานะสำหรับระบบที่มีขนาดใหญ่ โดยการวิเคราะห์ข้อมูลผลลัพธ์ที่ได้จากซิมมอยด์ฟังก์ชันเพิ่มเติมหรือเพิ่มขีดแบ่งตามก่อนข้อมูล เช่น อาจเพิ่มขีดแบ่งให้มีมากกว่า 1 ขีดแบ่ง เพื่อให้ในการเลือกผลลัพธ์ที่แท้จริงสำหรับข้อมูลแต่ละก้อน

2) ปรับปรุงอัลกอริทึมการเตรียมข้อมูลให้มีความสามารถในการรองรับแบบจำลองเชิงรูปนัยแบบนำเข้าแบบอื่น เช่น ไทม์ออโตมาตา (Time automata) หรือ โครงสร้างคริปกี (Kripke structure) เป็นต้น



## บรรณานุกรม

1. Balci, O. *Verification validation and accreditation of simulation models*. in *Proceedings of the 29th conference on Winter simulation*. 1997.
2. Manna, Z. and A. Pnueli, *Axiomatic approach to total correctness of programs*. Acta Informatica, 1974. **3**: p. 243-263.
3. Clarke, E.M. and E.A. Emerson. *Design and synthesis of synchronization skeletons using branching time temporal logic*. in *Workshop on logic of programs*. 1981. Springer.
4. Pelánek, R., *Reduction and abstraction techniques for model checking*. 2006, Masarykova univerzita, Fakulta informatiky.
5. Hochreiter, S. and J. Schmidhuber, *Long short-term memory*. Neural computation, 1997. **9**(8): p. 1735-1780.
6. Song, S., H. Huang, and T. Ruan, *Abstractive text summarization using LSTM-CNN based deep learning*. Multimedia Tools and Applications, 2019. **78**(1): p. 857-875.
7. Yang, Z. and G. Liu, *Hierarchical sequence-to-sequence model for multi-label text classification*. IEEE Access, 2019. **7**: p. 153012-153020.
8. Yang, P., et al., *SGM: sequence generation model for multi-label classification*. arXiv preprint arXiv:1806.04822, 2018.
9. Wang, Z., J. Poon, and S. Poon. *Tcm translator: A sequence generation approach for prescribing herbal medicines*. in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2019. IEEE.
10. Chen, G., et al. *Ensemble application of convolutional and recurrent neural networks for multi-label text categorization*. in *2017 International joint conference on neural networks (IJCNN)*. 2017. IEEE.
11. Ma, H., et al., *Mscoc: Multi-step co-attention model for multi-label classification*. IEEE Access, 2019. **7**: p. 109635-109645.
12. Lipton, Z.C., D.C. Kale, C. Elkan, and R. Wetzel, *Learning to diagnose with LSTM recurrent neural networks*. arXiv preprint arXiv:1511.03677, 2015.
13. Read, J., B. Pfahringer, and G. Holmes. *Multi-label classification using ensembles of pruned sets*. in *2008 eighth IEEE international conference on data mining*. 2008. IEEE.
14. Hua, Y., L. Mou, and X.X. Zhu, *Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional LSTM network for multi-label aerial image classification*. ISPRS journal of photogrammetry and remote sensing, 2019. **149**: p. 188-199.
15. Chen, Z. and J. Ren, *Multi-label text classification with latent word-wise label information*.

- Applied Intelligence, 2021. **51**(2): p. 966-979.
16. Al-Otaibi, R., P. Flach, and M. Kull. *Multi-label classification: A comparative study on threshold selection methods*. in *First International Workshop on Learning over Multiple Contexts (LMCE) at ECML-PKDD*. 2014.
  17. Ben-Ari, M., *Principles of the Spin model checker*. 2008: Springer Science & Business Media.
  18. *Advanced Topics in Promela*, in *Principles of the Spin Model Checker*. 2008, Springer London: London. p. 137-144.
  19. Berthomieu\*, B., P.-O. Ribet, and F. Vernadat, *The tool TINA—construction of abstract state spaces for Petri nets and time Petri nets*. International journal of production research, 2004. **42**(14): p. 2741-2756.
  20. Jensen, K., L.M. Kristensen, and L. Wells, *Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems*. International Journal on Software Tools for Technology Transfer, 2007. **9**: p. 213-254.
  21. Choi, B.W., *Petri net approaches for modeling, controlling, and validating flexible manufacturing systems*. 1994, Iowa State University.
  22. Reisig, W., *Understanding petri nets: modeling techniques, analysis methods, case studies*. 2013: Springer.
  23. Kalman, R.E., *A new approach to linear filtering and prediction problems*. 1960.
  24. Stock, J.H. and M.W. Watson, *Dynamic factor models, factor-augmented vector autoregressions, and structural vector autoregressions in macroeconomics*, in *Handbook of macroeconomics*. 2016, Elsevier. p. 415-525.
  25. Roesser, R., *A discrete state-space model for linear image processing*. IEEE Transactions on Automatic Control, 1975. **20**(1): p. 1-10.
  26. Pnueli, A., *The temporal logic of programs*. 18th Annual Symposium on Foundations of Computer Science (sfcs 1977), 1977: p. 46-57.
  27. Fisher, M., *An introduction to practical formal methods using temporal logic*. 2011: John Wiley & Sons.
  28. Lamport, L., *Proving the correctness of multiprocess programs*. IEEE transactions on software engineering, 1977(2): p. 125-143.
  29. Baier, C. and J.-P. Katoen, *Principles of model checking*. 2008: MIT press.
  30. Graves, A., A.-r. Mohamed, and G. Hinton. *Speech recognition with deep recurrent neural networks*. in *2013 IEEE international conference on acoustics, speech and signal processing*. 2013. IEEE.
  31. Yao, L. and Y. Guan. *An improved LSTM structure for natural language processing*. in *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*. 2018. IEEE.

32. Bengio, Y., P. Simard, and P. Frasconi, *Learning long-term dependencies with gradient descent is difficult*. IEEE transactions on neural networks, 1994. **5**(2): p. 157-166.
33. Olah, C., *Understanding lstm networks*. 2015.
34. de Carvalho, A.C. and A.A. Freitas, *A tutorial on multi-label classification techniques*. Foundations of Computational Intelligence Volume 5: Function Approximation and Classification, 2009: p. 177-195.
35. Godbole, S. and S. Sarawagi. *Discriminative methods for multi-labeled classification*. in *Pacific-Asia conference on knowledge discovery and data mining*. 2004. Springer.
36. Schapire, R.E. and Y. Singer. *Improved boosting algorithms using confidence-rated predictions*. in *Proceedings of the eleventh annual conference on Computational learning theory*. 1998.
37. Langeron, C., C. Moulin, and M. Géry. *MCut: A thresholding strategy for multi-label classification*. in *International Symposium on Intelligent Data Analysis*. 2012. Springer.
38. Yang, Y. *A study of thresholding strategies for text categorization*. in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. 2001.
39. Youden, W.J., *Index for rating diagnostic tests*. Cancer, 1950. **3**(1): p. 32-35.
40. Hanley, J.A. and B.J. McNeil, *The meaning and use of the area under a receiver operating characteristic (ROC) curve*. Radiology, 1982. **143**(1): p. 29-36.
41. Godefroid, P. *Model checking for programming languages using VeriSoft*. in *Proceedings of the 24th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. 1997.
42. Godefroid, P. and S. Khurshid. *Exploring very large state spaces using genetic algorithms*. in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. 2002. Springer.
43. Bošnački, D. *A light-weight algorithm for model checking with symmetry reduction and weak fairness*. in *International SPIN Workshop on Model Checking of Software*. 2003. Springer.
44. Behrmann, G., K.G. Larsen, and R. Pelánek. *To store or not to store*. in *Computer Aided Verification: 15th International Conference, CAV 2003, Boulder, CO, USA, July 8-12, 2003. Proceedings 15*. 2003. Springer.
45. Turner, E., M. Leuschel, C. Spermann, and M. Butler. *Symmetry reduced model checking for B*. in *First Joint IEEE/IFIP Symposium on Theoretical Aspects of Software Engineering (TASE'07)*. 2007. IEEE.
46. Alba, E., F. Chicano, M. Ferreira, and J. Gomez-Pulido. *Finding deadlocks in large concurrent java programs using genetic algorithms*. in *Proceedings of the 10th annual conference on*

- Genetic and evolutionary computation*. 2008.
47. Chicano, F. and E. Alba. *Finding liveness errors with ACO*. in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. 2008. IEEE.
  48. Araragi, T. and S.M. Cho. *Checking liveness properties of concurrent systems by reinforcement learning*. in *International Workshop on Model Checking and Artificial Intelligence*. 2006. Springer.
  49. Behjati, R., M. Sirjani, and M. Nili Ahmadabadi. *Bounded rational search for on-the-fly model checking of LTL properties*. in *Fundamentals of Software Engineering: Third IPM International Conference, FSEN 2009, Kish Island, Iran, April 15-17, 2009, Revised Selected Papers 3*. 2010. Springer.
  50. Bortolussi, L. and S. Silveti. *Bayesian statistical parameter synthesis for linear temporal properties of stochastic models*. in *Tools and Algorithms for the Construction and Analysis of Systems: 24th International Conference, TACAS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings, Part II 24*. 2018. Springer.
  51. Yasrebi, M., V. Rafe, and S. Nejatian, *An efficient approach to state space management in model checking of complex software systems using machine learning techniques*. *Journal of Intelligent & Fuzzy Systems*, 2020. **38**(2): p. 1761-1773.
  52. Partabian, J., V. Rafe, H. Parvin, and S. Nejatian, *An approach based on knowledge exploration for state space management in checking reachability of complex software systems*. *soft computing*, 2020. **24**: p. 7181-7196.
  53. Mukherjee, P. and T. Rompf, *A GNN Based Approach to LTL Model Checking*. arXiv preprint arXiv:2110.14824, 2021.
  54. Mehrabi, M.J. and V. Rafe, *Using deep reinforcement learning to search reachability properties in systems specified through graph transformation*. *Soft Computing*, 2022. **26**(18): p. 9635-9663.
  55. Zhu, W., H. Wu, and M. Deng, *LTL model checking based on binary classification of machine learning*. *IEEE access*, 2019. **7**: p. 135703-135719.
  56. Kripke, S., *Semantical Considerations on Modal Logic*. *Acta Philosophica Fennica*, 1963. **16**: p. 83-94.
  57. Zhu, W. and H. Wu, *CTL model checking based on binary classification of machine learning*. *Int. Arab J. Inf. Technol.*, 2022. **19**(2): p. 249-260.
  58. Wang, Q., et al. *Algorithm selection for software verification based on adversarial LSTM*. in *2021 7th IEEE Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on*

- Intelligent Data and Security (IDS)*. 2021. IEEE.
59. Famili, A., W.-M. Shen, R. Weber, and E. Simoudis, *Data preprocessing and intelligent data analysis*. *Intelligent data analysis*, 1997. **1**(1): p. 3-23.
  60. Liu, C., Z. Jin, J. Gu, and C. Qiu. *Short-term load forecasting using a long short-term memory network*. in *2017 IEEE PES innovative smart grid technologies conference Europe (ISGT-Europe)*. 2017. IEEE.
  61. Kaplan, R.M., *A method for tokenizing text*. *Inquiries into words, constraints and contexts*, 2005. **55**.
  62. Kingma, D.P. and J. Ba, *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980, 2014.
  63. Van Rossum, G. and F.L. Drake, *Introduction to python 3: python documentation manual part 1*. 2009: CreateSpace.
  64. Lundh, F., *An introduction to tkinter*. URL: [www.pythonware.com/library/tkinter/introduction/index.htm](http://www.pythonware.com/library/tkinter/introduction/index.htm), 1999.
  65. Abadi, M., et al. *{TensorFlow}: a system for {Large-Scale} machine learning*. in *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. 2016.
  66. Chollet, F., *Keras.[online]* Available at: <https://github.com/fchollet/keras>. Accessed, 2015. **12**(01): p. 2021.
  67. Tanenbaum, A.S. and A.S. Woodhull, *Operating systems: design and implementation*. Vol. 68. 1997: Prentice Hall Englewood Cliffs.
  68. Kordon, F., et al., *Complete Results for the 2020 Edition of the Model Checking Contest; 2021*.
  69. Boutell, M.R., J. Luo, X. Shen, and C.M. Brown, *Learning multi-label scene classification*. *Pattern recognition*, 2004. **37**(9): p. 1757-1771.
  70. Zhang, M.-L. and Z.-H. Zhou, *ML-KNN: A lazy learning approach to multi-label learning*. *Pattern recognition*, 2007. **40**(7): p. 2038-2048.
  71. Spyromitros, E., G. Tsoumakas, and I. Vlahavas. *An empirical study of lazy multilabel classification algorithms*. in *Artificial Intelligence: Theories, Models and Applications: 5th Hellenic Conference on AI, SETN 2008, Syros, Greece, October 2-4, 2008. Proceedings 5*. 2008. Springer.

ภาคผนวก ก.

คำอธิบายยูสเคส

ในบทนี้ ผู้วิจัยได้เขียนคำอธิบายยูสเคสที่แสดงอยู่ในรูปที่ 5.1 โดยมีรายละเอียด ดังต่อไปนี้

ตารางที่ ก.1 คำอธิบายยูสเคส Browse directory

หมายเลขยูสเคส : UC01	ชื่อยูสเคส : Browse directory
คำอธิบายยูสเคส : อนุญาตให้ผู้ใช้เรียกดูและเลือกไต่แรกทอรีของเครื่องสร้างปฏิภูมิสถานะ สำหรับระบบที่สนใจ	
ผู้เกี่ยวข้องหลัก : ผู้ใช้	
สิ่งกระตุ้น : ผู้ใช้คลิกปุ่ม “Browse”	
เงื่อนไขก่อน : -	
ความสัมพันธ์ : -	
กระแสเหตุการณ์ปกติ: <ol style="list-style-type: none"><li>1. ผู้ใช้คลิกปุ่ม “Browse” ในการกำหนดไต่แรกทอรีของเครื่องสร้างปฏิภูมิสถานะสำหรับระบบที่สนใจ</li><li>2. ระบบเปิดหน้าต่างสำหรับเลือกไต่แรกทอรี</li><li>3. ผู้ใช้เลือกไต่แรกทอรี</li><li>4. เส้นทางไต่แรกทอรีที่เลือกแสดงในกล่องข้อความ</li></ol>	
กระแสเหตุการณ์ทางเลือก: <ol style="list-style-type: none"><li>1. ยกเลิกการเลือกไต่แรกทอรี</li><li>2. หน้าต่างสำหรับเลือกไต่แรกทอรีถูกปิด</li><li>3. เส้นทางไต่แรกทอรีที่ไม่ถูกแสดงในกล่องข้อความ</li></ol>	
เงื่อนไขหลัง : -	

ตารางที่ ก.2 คำอธิบายยูสเคส Generate state space

หมายเลขยูสเคส : UC02	ชื่อยูสเคส : Generate state space
คำอธิบายยูสเคส : สร้างปริภูมิสถานะของระบบที่สนใจ	
ผู้เกี่ยวข้องหลัก : ผู้ใช้	
สิ่งกระตุ้น : ผู้ใช้คลิกปุ่ม “Generate State Space”	
เงื่อนไขก่อน : <ol style="list-style-type: none"> <li>1. กำหนดเส้นทางไคเรททอรีของเครื่องสร้างปริภูมิสถานะสำหรับระบบที่สนใจ</li> <li>2. ข้อมูลที่ผู้ใช้ป้อนเข้ามาได้รับการตรวจสอบ</li> <li>3. สถานะของระบบอยู่ในสถานะพร้อม</li> </ol>	
ความสัมพันธ์ : <p>Include: Validate User Input</p>	
กระแสเหตุการณ์ปกติ: <ol style="list-style-type: none"> <li>1. เลือกเส้นทางไคเรททอรีของเครื่องสร้างปริภูมิสถานะ</li> <li>2. ผู้ใช้กำหนดเส้นทางดำเนินงานและกำหนดความยาวของเส้นทางดำเนินงานที่ต้องการสร้าง</li> <li>2. คลิกปุ่ม “Generate State Space”</li> <li>4. สถานะของระบบอยู่ในสถานะกำลังสร้าง</li> <li>5. ระบบตรวจสอบการป้อนข้อมูลของผู้ใช้ คือ เส้นทางไคเรททอรีของเครื่องสร้างปริภูมิสถานะ เส้นทางดำเนินงานและกำหนดค่าการวนซ้ำ</li> <li>6. สร้างปริภูมิสถานะที่อยู่ในรูปแบบของเส้นทางดำเนินงาน</li> <li>7. แสดงปริภูมิสถานะที่อยู่ในรูปแบบของเส้นทางดำเนินงาน</li> <li>8. สถานะของระบบอยู่ในสถานะเสร็จเรียบร้อย</li> </ol>	
กระแสเหตุการณ์ทางเลือก: <ol style="list-style-type: none"> <li>3a. หากไม่ได้ระบุเส้นทางไคเรททอรีของเครื่องสร้างปริภูมิสถานะ หรือ ป้อนเส้นทางดำเนินงาน <ol style="list-style-type: none"> <li>3a1. แสดงหน้าต่างคำเตือนให้กำหนดใหม่อีกครั้ง</li> </ol> </li> <li>3b. ไม่พบเครื่องสร้างปริภูมิสถานะในเส้นทางไคเรททอรีที่กำหนด <ol style="list-style-type: none"> <li>3b1. แสดงคำเตือนไม่พบเครื่องสร้างปริภูมิสถานะ</li> </ol> </li> </ol>	
เงื่อนไขหลัง:	

ตารางที่ ก.3 คำอธิบายยูสเคส Validate user input

หมายเลขยูสเคส : UC03	ชื่อยูสเคส : Validate user input
คำอธิบายยูสเคส : ตรวจสอบการป้อนข้อมูลของผู้ใช้เพื่อความถูกต้องและครบถ้วน	
ผู้เกี่ยวข้องหลัก : ผู้ใช้	
สิ่งกระตุ้น : ผู้ใช้คลิกปุ่ม “Generate State Space”	
เงื่อนไขก่อน : -	
ความสัมพันธ์ : -	
<b>กระแสเหตุการณ์ปกติ:</b> <ol style="list-style-type: none"> <li>1. ตรวจสอบเส้นทางไต่แรกทอรีสำหรับเครื่องสร้างปริภูมิสถานะจากข้อมูลที่ผู้ใช้ป้อนเข้ามา</li> <li>2. ตรวจสอบเส้นทางดำเนินงานที่ผู้ใช้ป้อน</li> <li>3. ตรวจสอบคุณสมบัติที่ต้องการทวนสอบ</li> <li>4. ตรวจสอบสูตรแอลทีแอลที่ผู้ใช้ป้อน</li> </ol>	
<b>กระแสเหตุการณ์ทางเลือก:</b> <p><b>สำหรับกระบวนการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติ</b></p> <ol style="list-style-type: none"> <li>1a. ไม่พบการกรอกกล่องข้อความไต่แรกทอรี <ol style="list-style-type: none"> <li>1a.1. แสดงหน้าต่างคำเตือนให้กำหนดไต่แรกทอรี</li> </ol> </li> <li>2a. ไม่พบเครื่องสร้างปริภูมิสถานะในเส้นทางไต่แรกทอรีที่กำหนด <ol style="list-style-type: none"> <li>2a.1. แสดงหน้าต่างคำเตือนให้กำหนดไต่แรกทอรีใหม่อีกครั้ง</li> </ol> </li> </ol> <p><b>สำหรับกระบวนการสร้างปริภูมิสถานะ</b></p> <ol style="list-style-type: none"> <li>3b. ไม่พบการกรอกเส้นทางดำเนินงาน สำหรับกระบวนการสร้างปริภูมิสถานะ <ol style="list-style-type: none"> <li>1b.1. แสดงหน้าต่างคำเตือนให้กรอกเส้นทางดำเนินงาน</li> </ol> </li> <li>3a. เส้นทางดำเนินงานที่กำหนดมีความยาวเกินกว่ากำหนด โดยค่าปริยายเท่ากับ 4 <ol style="list-style-type: none"> <li>3a.1. แสดงหน้าต่างคำเตือนเส้นทางดำเนินงานมีความยาวเกินกว่ากำหนด</li> </ol> </li> </ol> <p><b>สำหรับกระบวนการทวนสอบคุณสมบัติการทำงานของระบบ</b></p> <ol style="list-style-type: none"> <li>4a. ผู้ใช้ไม่เลือกคุณสมบัติที่ต้องการทวนสอบ <ol style="list-style-type: none"> <li>4a.1. แสดงหน้าต่างคำเตือนให้เลือกคุณสมบัติที่ต้องการทวนสอบ</li> </ol> </li> <li>5a. รูปแบบสูตรแอลทีแอลที่ผู้ใช้ป้อนไม่ถูกต้องตามคุณสมบัติที่เลือก <ol style="list-style-type: none"> <li>5a.1. แสดงหน้าต่างคำเตือนให้ป้อนสูตรแอลทีแอลใหม่อีกครั้ง</li> </ol> </li> </ol>	
เงื่อนไขหลัง: -	



ตารางที่ ก.4 คำอธิบายยูสเคส Show generate state space

หมายเลขยูสเคส : UC04	ชื่อยูสเคส : Show generate state space
คำอธิบายยูสเคส : แสดงปริภูมิสถานะที่สร้างขึ้น โดยอยู่ในรูปแบบของเส้นทางดำเนินงาน	
ผู้เกี่ยวข้องหลัก : ผู้ใช้	
สิ่งกระตุ้น : -	
เงื่อนไขก่อน : <ol style="list-style-type: none"> <li>1. ผลการตรวจสอบข้อมูลที่ผู้ใช้ป้อนมีความถูกต้องและสมบูรณ์</li> <li>2. สร้างปริภูมิสถานะถูกสร้างเรียบร้อย</li> </ol>	
ความสัมพันธ์ : Include: Generate State Space	
กระแสเหตุการณ์ปกติ: <ol style="list-style-type: none"> <li>1. แสดงปริภูมิสถานะที่ถูกสร้างขึ้น ซึ่งอยู่ในรูปแบบของเส้นทางดำเนินงาน</li> </ol>	
กระแสเหตุการณ์ทางเลือก: -	
เงื่อนไขหลัง: -	

ตารางที่ ก.5 คำอธิบายยูสเคส Export generated state space

หมายเลขยูสเคส : UC05	ชื่อยูสเคส : Export generated state space
คำอธิบายยูสเคส : ส่งออกปริภูมิสถานะที่สร้างขึ้น ที่อยู่ในรูปแบบของเส้นทางดำเนินงาน	
ผู้เกี่ยวข้องหลัก : ผู้ใช้	
สิ่งกระตุ้น : ผู้ใช้คลิกปุ่ม “Export result”	
เงื่อนไขก่อน : <ol style="list-style-type: none"> <li>1. ผลการตรวจสอบข้อมูลที่ใช้ป้อนมีความถูกต้องและสมบูรณ์</li> <li>2. สร้างปริภูมิสถานะถูกสร้างเรียบร้อยแล้ว</li> <li>3. ปริภูมิสถานะที่ถูกสร้างขึ้น ถูกแสดงในหน้าจอ</li> </ol>	
ความสัมพันธ์ : <p>Include: Show Generate State Space</p>	
กระแสเหตุการณ์ปกติ: <ol style="list-style-type: none"> <li>1. สร้างไฟล์นามสกุล “.csv”</li> <li>2. บันทึกเส้นทางดำเนินงานในไฟล์ “.csv” ที่สร้าง</li> <li>3. แสดงหน้าต่างการส่งออกผลลัพธ์ถูกสร้างเรียบร้อยแล้ว</li> </ol>	
กระแสเหตุการณ์ทางเลือก: <ol style="list-style-type: none"> <li>3a. ผู้ใช้เลือกเปิดไฟล์ของผลลัพธ์ที่ถูกส่งออก <ol style="list-style-type: none"> <li>3a.1. เปิดไฟล์ที่ถูกส่งออก</li> </ol> </li> <li>3b. ผู้ใช้ไม่เลือกเปิดไฟล์ของผลลัพธ์ที่ถูกส่งออก <ol style="list-style-type: none"> <li>3b.1 ปิดหน้าต่างการแสดงผลการส่งออก</li> </ol> </li> </ol>	
เงื่อนไขหลัง: -	

ตารางที่ ก.6 คำอธิบายยูสเคส Verify LTL property

หมายเลขยูสเคส : UC06	ชื่อยูสเคส : Verify LTL property
คำอธิบายยูสเคส : การทวนสอบคุณสมบัติการทำงานของระบบที่สนใจ	
ผู้เกี่ยวข้องหลัก : ผู้ใช้	
สิ่งกระตุ้น : ผู้ใช้คลิกปุ่ม “Verify”	
เงื่อนไขก่อน : <ol style="list-style-type: none"> <li>1. กำหนดเส้นทางไตรรกทอริของเครื่องสร้างปริภูมิสถานะสำหรับระบบที่สนใจ</li> <li>2. ข้อมูลที่ผู้ใช้ป้อนเข้ามาได้รับการตรวจสอบ</li> <li>3. ข้อมูลที่กำหนดได้รับการตรวจสอบความถูกต้อง</li> <li>4. ระบบอยู่ในสถานะพร้อมใช้</li> </ol>	
ความสัมพันธ์ : <p>Include: Generate State space</p>	
กระแสเหตุการณ์ปกติ: <ol style="list-style-type: none"> <li>1. กำหนดเส้นทางไตรรกทอริของเครื่องสร้างปริภูมิสถานะ</li> <li>2. เลือกคุณสมบัติที่ต้องการทวนสอบ</li> <li>3. ผู้ใช้กำหนดสูตรแอลทีแอลโดยเลือกวิธีการอย่างใดอย่างหนึ่ง คือ <ol style="list-style-type: none"> <li>3.1 ผู้ใช้ป้อนสูตรแอลทีแอลที่ต้องและเลือกคุณสมบัติการทำงานที่ต้องการทวนสอบ</li> <li>3.2 ผู้ใช้นำเข้าไฟล์สำหรับสูตรแอลทีแอล นามสกุล “.csv”</li> </ol> </li> <li>4. กำหนดความลึกที่ต้องการทวนสอบ</li> <li>5. ตรวจสอบความถูกต้องของข้อมูลที่ถูกป้อนทั้งหมด</li> <li>6. ทวนสอบคุณสมบัติการทำงานของระบบ</li> <li>7. ระบบอยู่ในสถานะกำลังทวนสอบ</li> <li>8. แสดงผลการทวนสอบคุณสมบัติการทำงานของระบบ</li> <li>9. ระบบอยู่ในสถานะเสร็จเรียบร้อยแล้ว</li> </ol>	
กระแสเหตุการณ์ทางเลือก: <ol style="list-style-type: none"> <li>4a. หากข้อมูลที่ผู้ใช้ป้อนไม่ถูกต้อง <ol style="list-style-type: none"> <li>4a.1 แสดงหน้าต่างคำเตือนให้กรอกข้อมูลใหม่</li> </ol> </li> </ol>	
เงื่อนไขหลัง: -	

ตารางที่ ก.7 คำอธิบายยูสเคส Fill LTL formula

หมายเลขยูสเคส : UC07	ชื่อยูสเคส : Fill LTL formula
คำอธิบายยูสเคส : อนุญาตให้ผู้ใช้ป้อนสูตรแอลทีแอลด้วยตัวเอง	
ผู้เกี่ยวข้องหลัก : ผู้ใช้	
สิ่งกระตุ้น : -	
เงื่อนไขก่อน : เลือกคุณสมบัติที่ต้องการทวนสอบ	
ความสัมพันธ์ : -	
กระแสเหตุการณ์ปกติ: 1. ผู้ใช้ป้อนสูตรแอลทีแอลด้วยตัวเองในกล่องข้อความที่กำหนดไว้	
กระแสเหตุการณ์ทางเลือก: -	
เงื่อนไขหลัง: -	

ตารางที่ ก.8 คำอธิบายยูสเคส Import LTL formula

หมายเลขยูสเคส : UC08	ชื่อยูสเคส : Import LTL formula
คำอธิบายยูสเคส : อนุญาตให้ผู้ใช้นำเข้าไฟล์สำหรับสูตรแอลทีแอล	
ผู้เกี่ยวข้องหลัก : ผู้ใช้	
สิ่งกระตุ้น : ผู้ใช้คลิกปุ่ม “Import LTL”	
เงื่อนไขก่อน : ไฟล์สำหรับสูตรแอลทีแอลที่ต้องการทวนสอบถูกสร้าง	
ความสัมพันธ์ : -	
กระแสเหตุการณ์ปกติ: 1. ผู้ใช้คลิกปุ่ม “Import LTL” 2. หน้าต่างสำหรับเลือกไฟล์นามสกุล “.csv” ถูกเปิด 3. ผู้ใช้เลือกไฟล์สำหรับสูตรแอลทีแอลที่ได้จัดเตรียมไว้ 4. กล่องข้อความของสูตรแอลทีแอลถูกระบุด้วยเส้นทางของไฟล์สำหรับสูตรแอลทีแอลที่โหลดเข้ามาในระบบ	
กระแสเหตุการณ์ทางเลือก: -	
เงื่อนไขหลัง: -	

ตารางที่ ก.9 คำอธิบายยูสเคส Show verification result

หมายเลขยูสเคส : UC09	ชื่อยูสเคส : Show verification result
คำอธิบายยูสเคส : แสดงผลลัพธ์การทวนสอบคุณสมบัติการทำงาน	
ผู้เกี่ยวข้องหลัก : ผู้ใช้	
สิ่งกระตุ้น :	
เงื่อนไขก่อน : เมื่อระบบทำการทวนสอบคุณสมบัติการทำงานเสร็จเรียบร้อยแล้ว	
ความสัมพันธ์ : Include: Verify LTL Property	
<b>กระแสเหตุการณ์ปกติ:</b> <ol style="list-style-type: none"> <li>1. แสดงผลการทวนสอบคุณสมบัติที่หน้าจอ</li> <li>2. สถานะของระบบเสร็จเรียบร้อยแล้ว</li> </ol>	
<b>กระแสเหตุการณ์ทางเลือก:</b> <ol style="list-style-type: none"> <li>1a. หากผู้ใช้อ้อนสูตรแอลทีแอลด้วยตัวเอง <ol style="list-style-type: none"> <li>1a.1 ผลลัพธ์การทวนสอบแสดงที่หน้าจอ</li> </ol> </li> <li>2b. หากผู้ใช้เลือกนำเข้าไฟล์สำหรับสูตรแอลทีแอล <ol style="list-style-type: none"> <li>2b.1 ผลลัพธ์การทวนสอบถูกออกรายงานเก็บไว้ในไดเรกทอรีที่เลือก</li> <li>2b.2 แสดงข้อความว่ารายงานผลลัพธ์การทวนสอบถูกจัดเก็บไว้ที่ใด ที่เดียวกับผลลัพธ์การทวนสอบแสดงที่หน้าจอ</li> </ol> </li> </ol>	
เงื่อนไขหลัง: -	

ตารางที่ ก.10 คำอธิบายยูสเคส Delete verification result

หมายเลขยูสเคส : UC010	ชื่อยูสเคส : Delete verification result
คำอธิบายยูสเคส : อนุญาตให้ผู้ใช้สามารถลบผลการทวนสอบได้	
ผู้เกี่ยวข้องหลัก : ผู้ใช้	
สิ่งกระตุ้น : ผู้ใช้คลิกปุ่ม “Delete result”	
เงื่อนไขก่อน : เมื่อระบบทำการทวนสอบคุณสมบัติการทำงานเสร็จเรียบร้อยแล้ว	
ความสัมพันธ์ : -	
กระแสเหตุการณ์ปกติ:	
<ol style="list-style-type: none"> <li>1. ผู้ใช้ โดยการคลิกปุ่ม “Delete result”</li> <li>2. ระบบลบผลลัพธ์การทวนสอบที่แสดงบนหน้าจอ</li> </ol>	
กระแสเหตุการณ์ทางเลือก: -	
เงื่อนไขหลัง: -	

ตารางที่ ก.11 คำอธิบายยูสเคส Show the total number of the generated exaction paths

หมายเลขยูสเคส : UC011	ชื่อยูสเคส : Show the total number of the generated exaction paths
คำอธิบายยูสเคส : แสดงจำนวนเส้นทางดำเนินงานทั้งหมดที่ถูกสร้างขึ้น	
ผู้เกี่ยวข้องหลัก : ผู้ใช้	
สิ่งกระตุ้น :	
เงื่อนไขก่อน :	
<ol style="list-style-type: none"> <li>1. ผลการตรวจสอบข้อมูลที่ใช้ป้อนมีความถูกต้องและสมบูรณ์</li> <li>2. สร้างปริภูมิสถานะถูกสร้างเรียบร้อยแล้ว</li> </ol>	
ความสัมพันธ์ : -	
กระแสเหตุการณ์ปกติ:	
<ol style="list-style-type: none"> <li>1. แสดงจำนวนเส้นทางดำเนินงานทั้งหมดที่ถูกสร้างออกมา บนหน้าจอ</li> </ol>	
กระแสเหตุการณ์ทางเลือก: -	
เงื่อนไขหลัง: -	

ตารางที่ ก.12 คำอธิบายยูสเคส Show example of specific LTL formulas

หมายเลขยูสเคส : UC012	ชื่อยูสเคส : Show example of specific LTL formulas
คำอธิบายยูสเคส : แสดงตัวอย่างสูตรแอลที่แอลแบบตามคุณสมบัติการทวนสอบ	
ผู้เกี่ยวข้องหลัก : ผู้ใช้	
สิ่งกระตุ้น : ผู้ใช้เลือกคุณสมบัติที่ต้องการทวนสอบบนหน้าจอ	
เงื่อนไขก่อน :	
ความสัมพันธ์ : -	
กระแสเหตุการณ์ปกติ: 1. แสดงตัวอย่างของสูตรแอลที่แอลตามคุณสมบัติที่ผู้ใช้เลือก	
กระแสเหตุการณ์ทางเลือก: -	
เงื่อนไขหลัง: -	

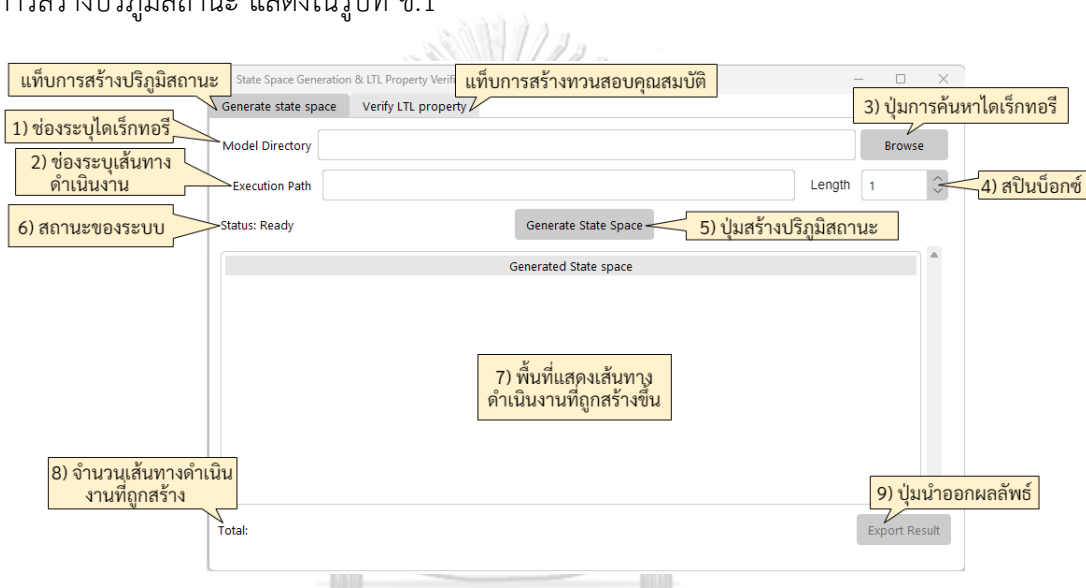
ตารางที่ ก.13 คำอธิบายยูสเคส Show status of the system

หมายเลขยูสเคส : UC013	ชื่อยูสเคส : Show status of the system
คำอธิบายยูสเคส : แสดงสถานะการทำงานของระบบ	
ผู้เกี่ยวข้องหลัก : ผู้ใช้	
สิ่งกระตุ้น : ผู้ใช้เปิดระบบขึ้นมาใช้งาน	
เงื่อนไขก่อน :	
ความสัมพันธ์ : -	
กระแสเหตุการณ์ปกติ: 1. แสดงสถานะของระบบบนหน้าจอ ได้แก่ สถานะพร้อมใช้ สถานะกำลังสร้าง สถานะกำลังทวนสอบและสถานะเสร็จเรียบร้อยแล้ว	
กระแสเหตุการณ์ทางเลือก: -	
เงื่อนไขหลัง: -	

## ภาคผนวก ข.

### การใช้งานระบบการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติ

เมื่อเรียกใช้ระบบการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติของระบบที่สนใจ ระบบแสดงหน้าต่างเริ่มต้น โดยแบ่งออกเป็น 2 ส่วนหลักการทำงานที่มีการทำงานที่อิสระต่อกัน คือ ส่วนการสร้างปริภูมิสถานะแสดงบนแท็บ “Generate state space” เมื่อผู้ใช้ต้องการสร้างปริภูมิสถานะ และส่วนของการทวนสอบคุณสมบัติแสดงบนแท็บ “Verify LTL property” เมื่อผู้ใช้ต้องการทวนสอบคุณสมบัติการทำงานของระบบที่สนใจ โดยการทำงานเริ่มต้นของระบบแสดงอยู่บนหน้าจอของการสร้างปริภูมิสถานะ แสดงในรูปที่ ข.1



รูปที่ ข.1 หน้าจอเริ่มต้นของระบบการสร้างปริภูมิสถานะและการทวนสอบคุณสมบัติ

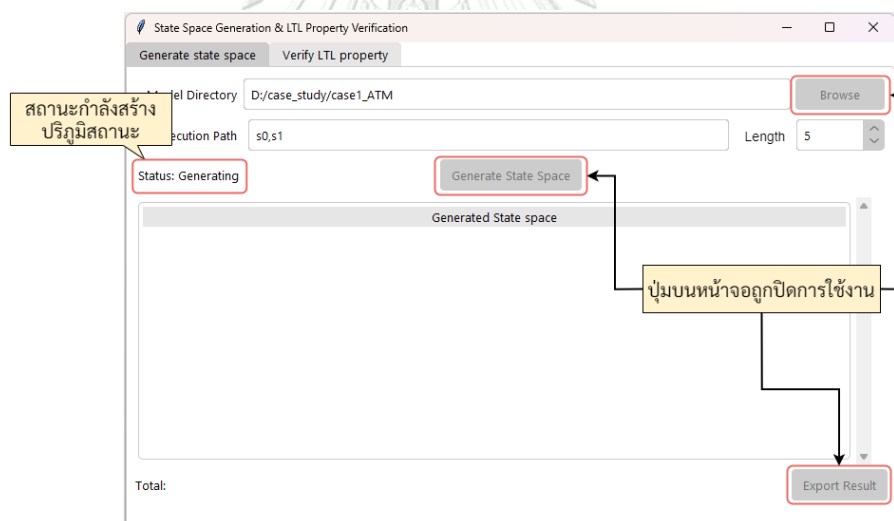
จากรูปที่ ข.1 ประกอบด้วย

- 1) ช่องข้อความสำหรับระบุไดเรกทอรีสำหรับเลือกแบบจำลอง ซึ่งสามารถค้นหาผ่านปุ่มค้นหา Browse
- 2) ช่องข้อความสำหรับระบุเส้นทางดำเนินงานเริ่มต้นสำหรับการสร้างปริภูมิสถานะ โดยมีเงื่อนไข คือ เส้นทางดำเนินงานจะต้องถูกระบุโดยใช้ชื่อสถานะเรียงต่อกันด้วยการใช้เครื่องหมายจุดภาคที่มีความยาวไม่เกิน 4
- 3) ปุ่มค้นหาไดเรกทอรี “Browse”
- 4) สปินบ็อกซ์ที่ใช้ระบุจำนวนความยาวของเส้นทางดำเนินงาน ซึ่งค่าโดยปริยายที่ 1
- 5) ปุ่มสำหรับสร้างปริภูมิสถานะ “Generate State Space”
- 6) ข้อความบอกสถานะการทำงานของระบบ โดยเริ่มต้นจะอยู่ในสถานะพร้อมใช้ (Ready)



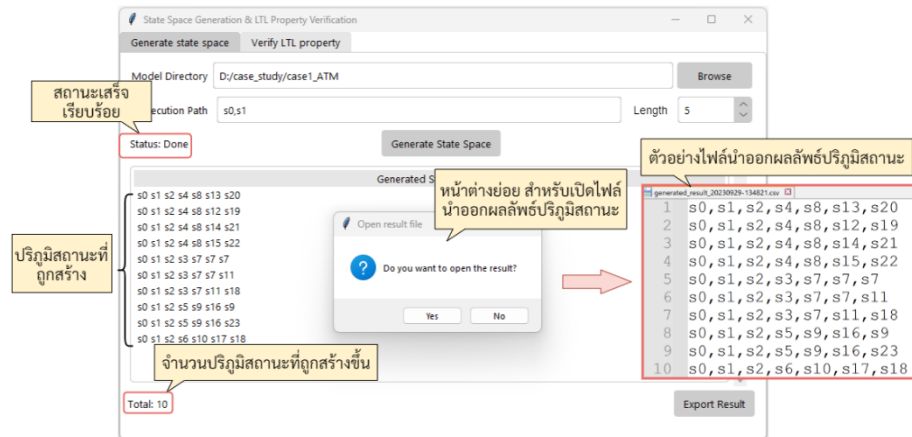
- 7) พื้นที่แสดงเส้นทางดำเนินงานที่ถูกสร้างขึ้น
- 8) ข้อความบอกจำนวนเส้นทางดำเนินงานที่ถูกสร้างทั้งหมด
- 9) ปุ่มนำออกผลลัพธ์การสร้างปริภูมิสถานะ “Export Result” ที่อยู่ในรูปแบบของไฟล์นามสกุล “.csv” ซึ่งปุ่มนำออกผลลัพธ์จะเปิดให้ใช้งานได้ก็ต่อเมื่อระบบสร้างปริภูมิสถานะเรียบร้อยแล้ว

เมื่อผู้ใช้ป้อนข้อมูลไตรรกทอรี เส้นทางดำเนินงานเริ่มต้น จำนวนความยาวของเส้นทางดำเนินงานที่ต้องการสร้างและผู้ใช้กดปุ่ม “Generate State Space” จากนั้น สถานะของระบบเปลี่ยนเป็นกำลังสร้าง (Generating) โดยที่ระบบเรียกใช้งานเครื่องสร้างปริภูมิสถานะที่อยู่ในไตรรกทอรีที่ระบุเข้ามาเพื่อทำการสร้างปริภูมิสถานะที่อยู่ในรูปแบบของเส้นทางดำเนินงานตามขนาดความยาวที่ผู้ใช้กำหนด ซึ่งในระหว่างที่ระบบกำลังทำงานสถานะของระบบเปลี่ยนเป็นกำลังสร้าง (Generating) และปุ่มบนหน้าจอถูกปิดการใช้งาน แสดงหน้าจอสำหรับระบบกำลังสร้างปริภูมิสถานะในรูปที่ ข.2



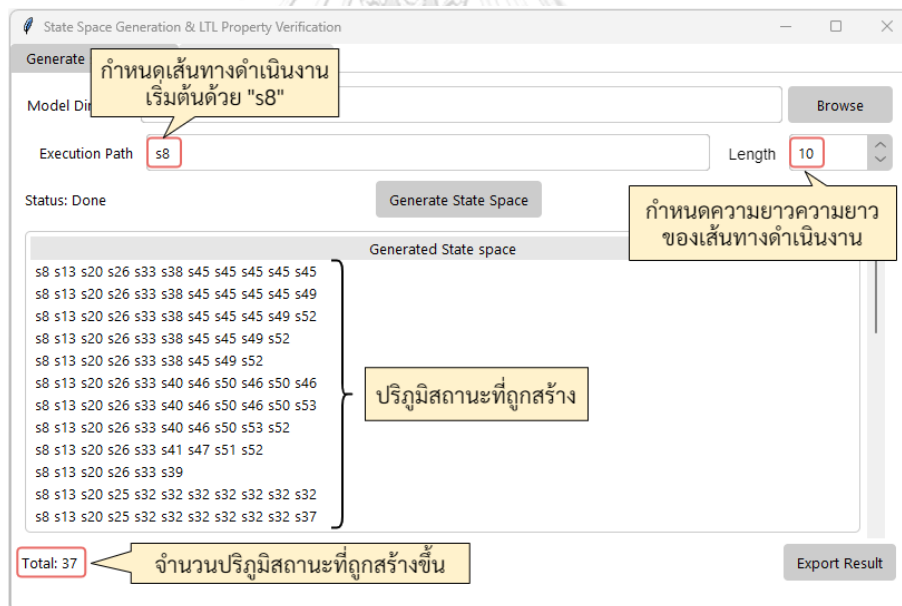
รูปที่ ข.2 หน้าจอสำหรับระบบกำลังสร้างปริภูมิสถานะ

เมื่อระบบสร้างปริภูมิสถานะเสร็จเรียบร้อยแล้ว สถานะของระบบเปลี่ยนเป็นเสร็จเรียบร้อยแล้ว (Done) ระบบแสดงปริภูมิสถานะที่สร้างขึ้นมาที่อยู่ในรูปแบบของเส้นทางดำเนินงานและแสดงจำนวนเส้นทางดำเนินงานทั้งหมดที่สร้างขึ้นมา จากนั้นปุ่มบนหน้าจอถูกเปิดใช้งานได้ กรณีที่ผู้ใช้เลือกนำออกผลลัพธ์ระบบดำเนินการออกผลลัพธ์ที่อยู่ในรูปแบบที่กำหนด เมื่อระบบนำออกผลลัพธ์เรียบร้อยแล้ว ระบบแสดงหน้าต่างย่อยสำหรับการเปิดไฟล์นำออกแบบอัตโนมัติ ในรูปที่ ข.3 แสดงหน้าจอเมื่อระบบสร้างปริภูมิสถานะเรียบร้อยแล้วและตัวอย่างไฟล์นำออกผลลัพธ์ปริภูมิสถานะ



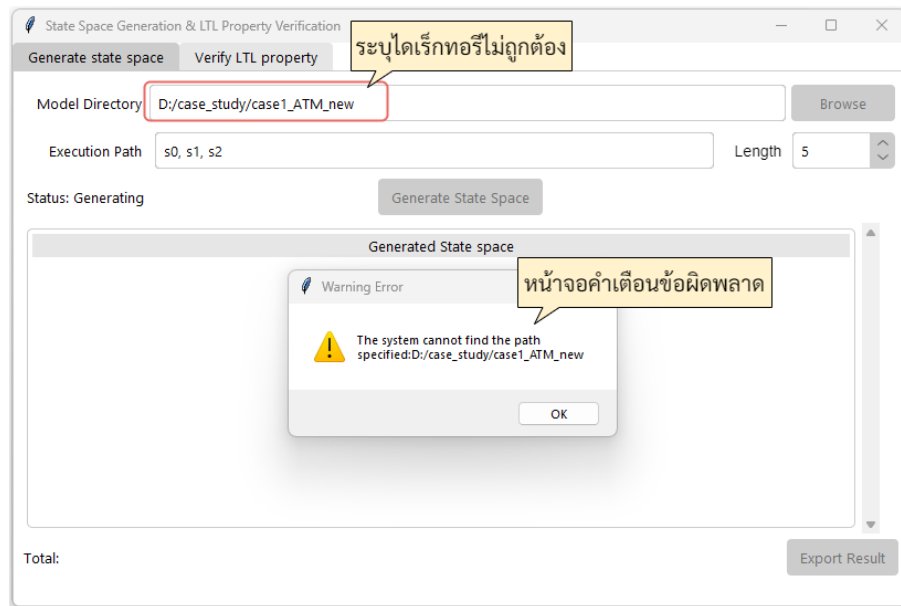
รูปที่ ข.3 หน้าจอแสดงผลการสร้างปริภูมิสถานะเรียบร้อยและตัวอย่างผลลัพธ์ปริภูมิสถานะ

อย่างไรก็ดี ระบบสามารถให้ผู้ใช้กำหนดเส้นทางดำเนินงานเริ่มต้นได้ตามต้องการ โดยไม่จำเป็นต้องเริ่มที่สถานะเริ่มต้นเสมอ ในรูปที่ ข.4 แสดงหน้าจอผลลัพธ์การสร้างปริภูมิสถานะจากเส้นทางดำเนินงานที่ไม่ได้เริ่มต้นจากสถานะเริ่มต้น



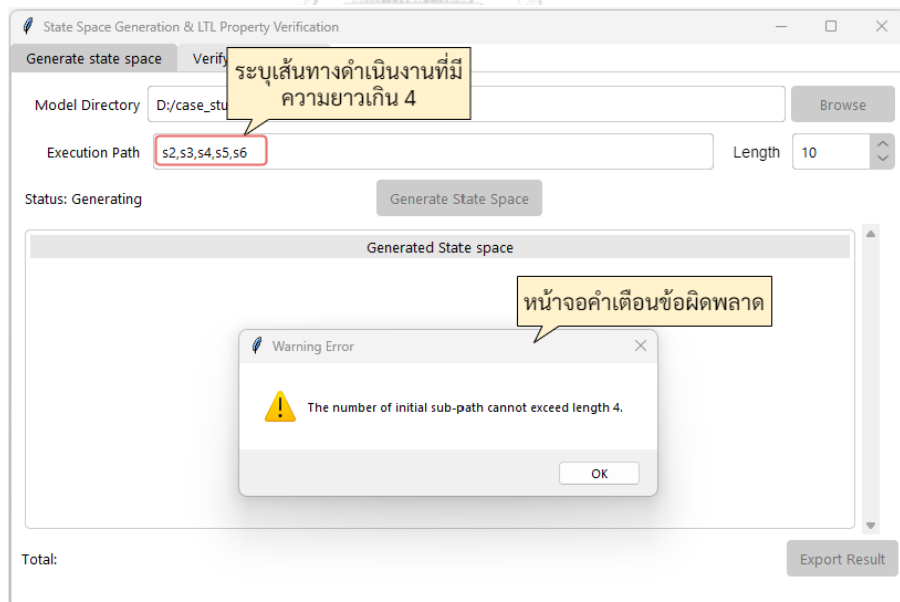
รูปที่ ข.4 หน้าจอแสดงผลการสร้างปริภูมิสถานะ

นอกจากนี้ ก่อนดำเนินการสร้างปริภูมิสถานะ ข้อมูลนำเข้าที่ระบุโดยผู้ใช้จะได้รับ การตรวจสอบความถูกต้องก่อนดำเนินการสร้างปริภูมิสถานะว่าเป็นไปตามข้อกำหนดหรือไม่ เช่น ไตรรกทอริไม่ถูกต้องหรือความยาวของเส้นทางดำเนินงานต้องมีขนาดไม่เกิน 4 เป็นต้น ในรูปที่ ข.5 ระบบแสดงหน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อผู้ใช้ระบุไตรรกทอริไม่ถูกต้อง



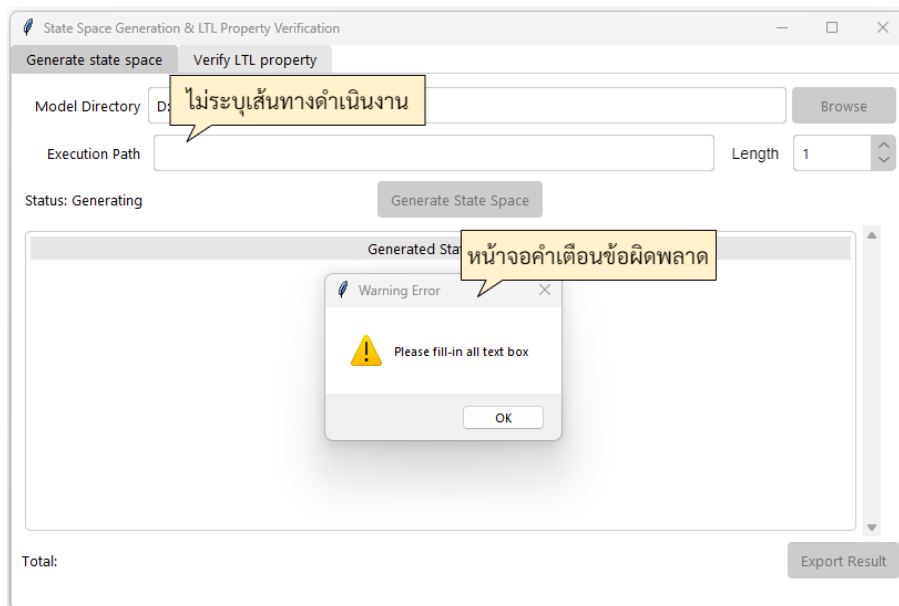
รูปที่ ข.5 หน้าจอแสดงหน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อผู้ใช้ระบุไดเรกทอรีไม่ถูกต้อง

ในรูปที่ ข.6 ระบบแสดงหน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อผู้ใช้ระบุเส้นทางดำเนินงานเริ่มต้นที่มีความยาวมากกว่า 4



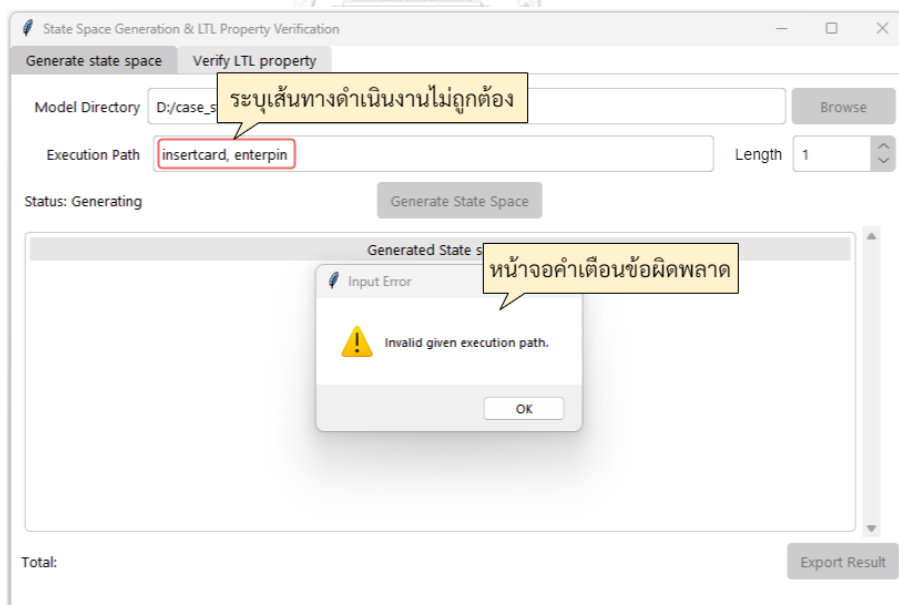
รูปที่ ข.6 หน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อกำหนดเส้นทางดำเนินงานมีความยาวเกินกว่าที่กำหนด

ในรูปที่ ข.7 ระบบแสดงหน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อผู้ใช้ไม่ระบุเส้นทางดำเนินงานเริ่มต้น



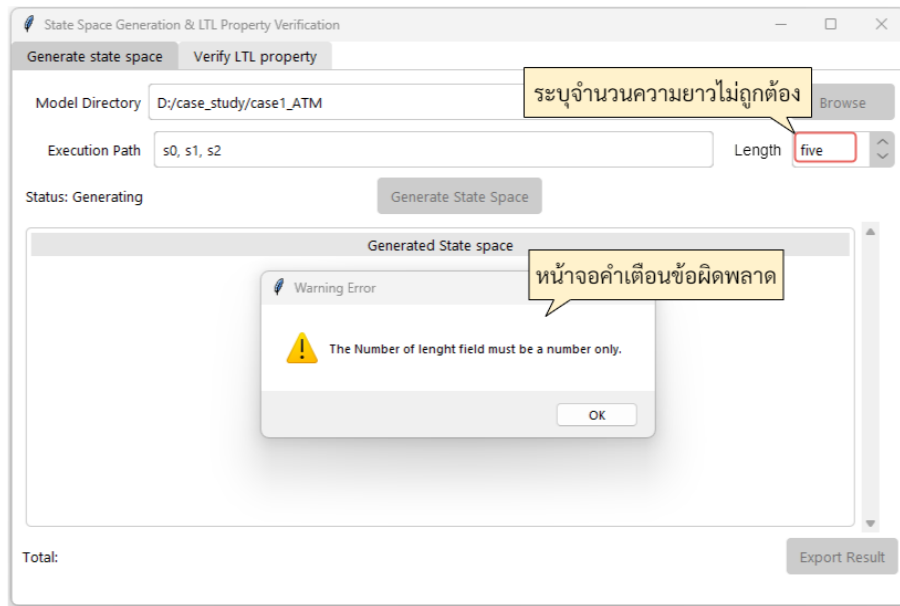
รูปที่ ข.7 หน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อผู้ใช้ไม่ระบุเส้นทางดำเนินงานเริ่มต้น

ในรูปที่ ข.8 ระบบแสดงหน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อผู้ใช้ระบุเส้นทางดำเนินงานโดยไม่ใช่ชื่อของสถานะ



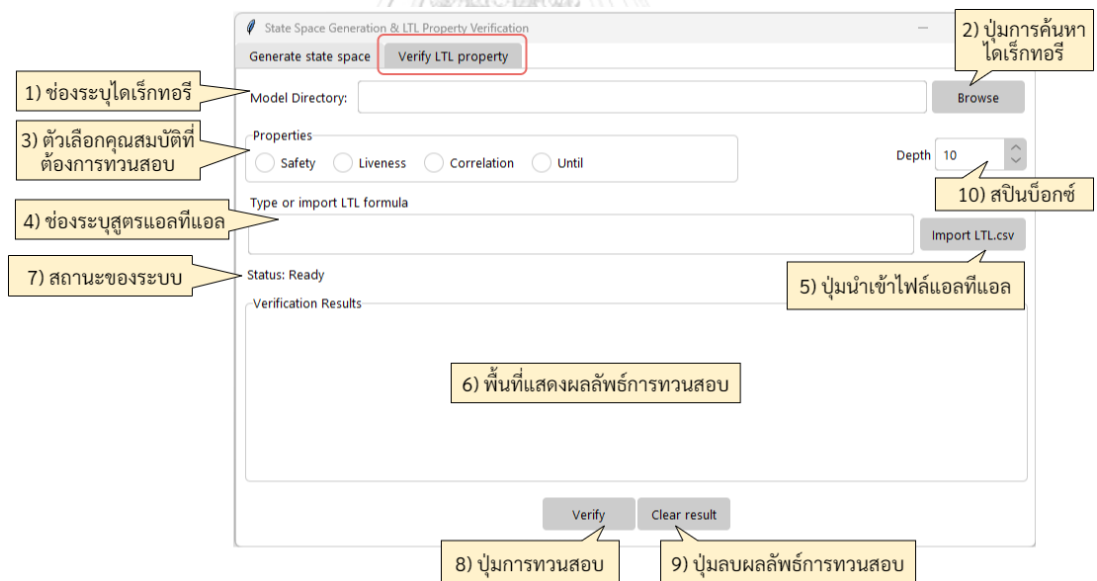
รูปที่ ข.8 หน้าต่างแจ้งเตือนข้อผิดพลาดผู้ใช้ระบุเส้นทางดำเนินงานเริ่มต้นไม่ถูกต้อง

ในรูปที่ ข.9 ระบบแสดงหน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อผู้ใช้ระบุจำนวนความยาวของเส้นทางดำเนินงานไม่ถูกต้อง



รูปที่ ข.9 หน้าต่างแจ้งเตือนข้อผิดพลาดผู้ใช้ระบุจำนวนความยาวไม่ถูกต้อง

สำหรับการทดสอบคุณสมบัติ “Verify LTL property” แสดงในรูปที่ ข.10



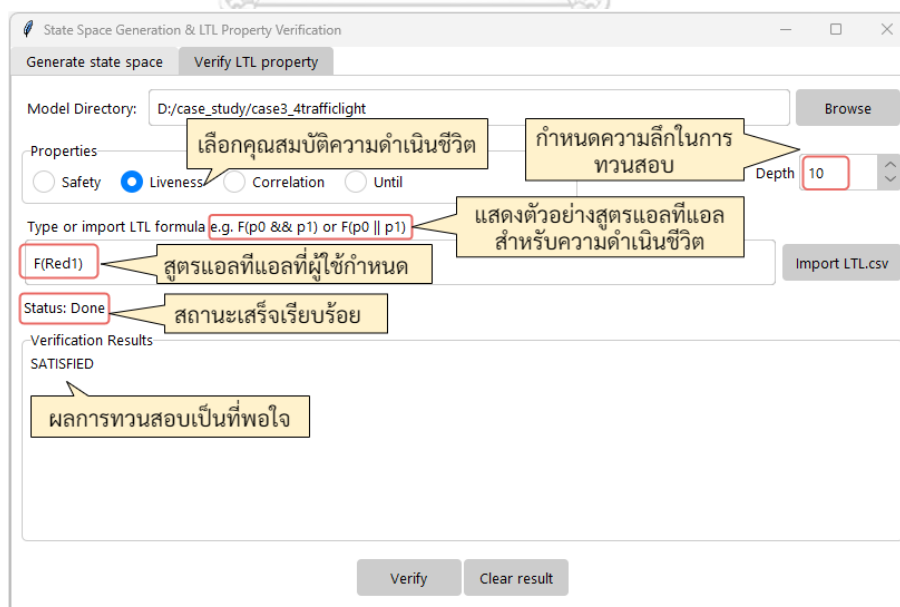
รูปที่ ข.10 หน้าจอแสดงแท็บการทดสอบคุณสมบัติ

ในรูปที่ ข.10 ประกอบด้วย

- 1) ช่องข้อความสำหรับระบุไดเรกทอรีสำหรับเลือกแบบจำลอง ซึ่งสามารถค้นหาผ่านปุ่มค้นหา Browse

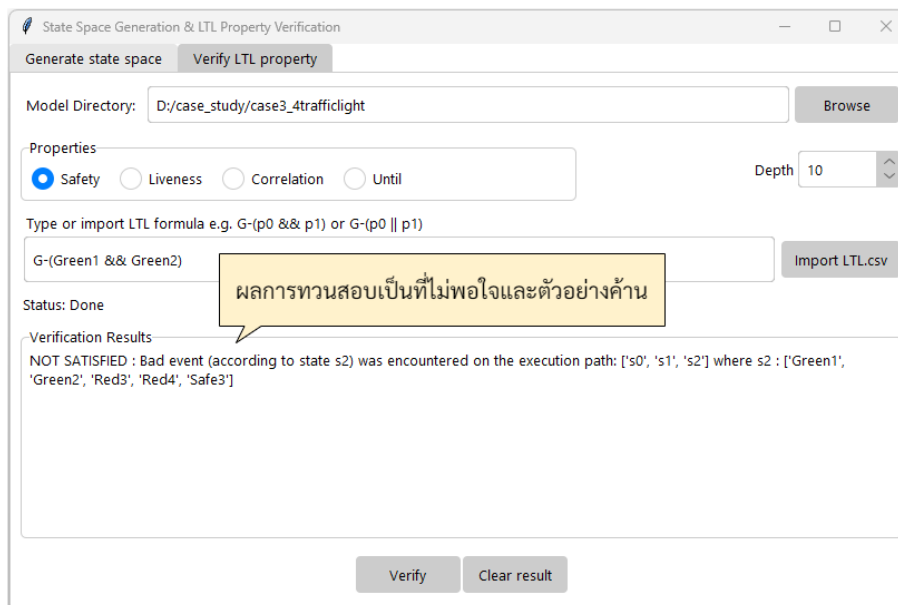
- 2) ปุ่มค้นหาไดเรกทอรี “Browse”
- 3) ตัวเลือกคุณสมบัติการทวนสอบ ได้แก่ ความปลอดภัย (Safety) ความดำเนินชีวิต (Liveness) ความสัมพันธ์ (Correlation) และสูตรแอลที่แอลประเภท Until
- 4) ช่องข้อความสำหรับระบุสูตรแอลที่แอลตามประเภทของคุณสมบัติที่เลือกหรือไดเรกทอรีของเอกสารนำเข้าสูตรแอลที่แอลที่เป็นไฟล์นามสกุล “.csv”
- 5) ปุ่มสำหรับนำเข้าเอกสารสูตรแอลที่แอล “Import LTL.csv”
- 6) พื้นที่แสดงผลลัพธ์การทวนสอบ
- 7) ข้อความบอกสถานการณ์ทำงานของระบบ โดยเริ่มต้นจะอยู่ในสถานะพร้อมใช้
- 8) ปุ่มการทวนสอบ “Verify”
- 9) ปุ่มลบผลลัพธ์การทวนสอบ “Clear result”
- 10) สปินบ็อกซ์ที่ใช้ระบุจำนวนความลึกของการทวนสอบ ซึ่งค่าโดยปริยายเริ่มต้นที่ 10

เมื่อผู้เลือกคุณสมบัติที่ต้องการทวนสอบ ระบบแสดงตัวอย่างของสูตรแอลที่แอลตามที่ใช้เลือก จากนั้นเมื่อผู้ใช้กดปุ่มการทวนสอบ “Verify” สถานะของระบบเปลี่ยนเป็นกำลังทวนสอบ (Verifying) โดยที่ระบบดำเนินการทวนสอบคุณสมบัติที่ผู้ใช้ป้อนเข้ามาตามความลึกการทวนสอบที่กำหนด เมื่อดำเนินการทวนสอบเรียบร้อย ระบบแสดงผลลัพธ์การทวนสอบที่พื้นที่การแสดงผลลัพธ์ และสถานะเปลี่ยนเป็นเสร็จเรียบร้อย และสถานะเปลี่ยนเป็นเสร็จเรียบร้อย แสดงในรูปที่ ข.11



รูปที่ ข.11 หน้าจอเมื่อระบบดำเนินการทวนสอบเสร็จเรียบร้อย

ในรูปที่ ข.12 แสดงกรณีที่ผลการทวนสอบเป็นที่ไม่พอใจ ระบบแสดงผลลัพธ์พร้อมทั้งตัวอย่างค่าน



รูปที่ ข.12 หน้าจอแสดงผลการทวนสอบเป็นที่ไม่พอใจ

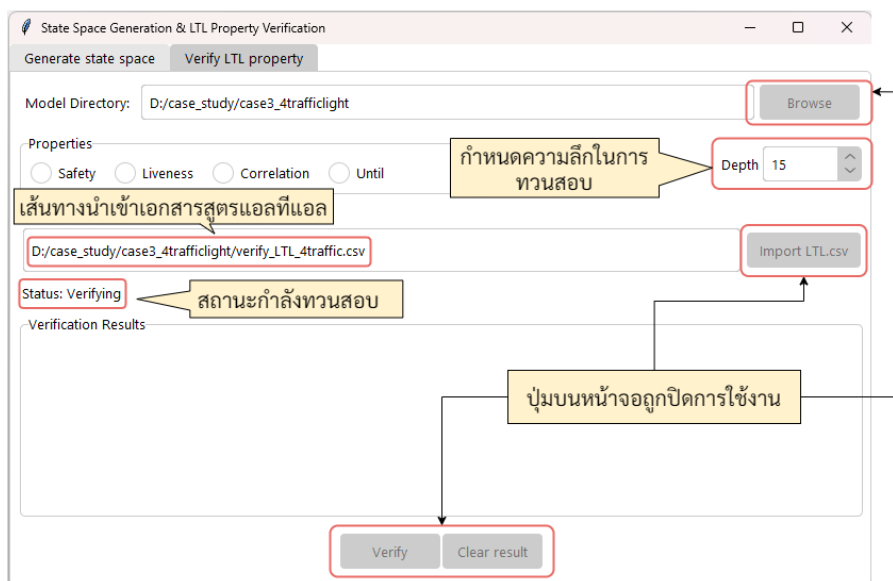
สำหรับการทวนสอบ ผู้ใช้สามารถนำเข้าเอกสารนามสกุล “.csv” ที่อยู่ในรูปแบบที่กำหนด แสดงในรูปที่ ข.13 ประกอบด้วย 6 คอลัมน์ ดังต่อไปนี้

- 1) คอลัมน์ “Test Case” ผู้ใช้ระบุลำดับของกรณีทดสอบ
- 2) คอลัมน์ “Property” ผู้ใช้ระบุคุณสมบัติที่ต้องการทวนสอบ
- 3) คอลัมน์ “Formula” ผู้ใช้ระบุสูตรแอลที่แอลที่สอดคล้องกับคุณสมบัติที่กำหนด
- 4) คอลัมน์ “Expected result” ผู้ใช้ระบุผลการทวนสอบที่คาดหวัง
- 5) คอลัมน์ “Actual Result” ระบบระบุผลการทวนสอบที่แท้จริง
- 6) คอลัมน์ “Comparison Result” ระบบระบุผลการเปรียบเทียบระหว่างผลการทวนสอบที่คาดหวังและผลการทวนสอบที่แท้จริง

Test Case	Property	Formula	Expected result	Actual Result	Comparison Result
FTL-TC-01	Safety	G-(Green1 && Green2 && Green3 && Green4)	NOT SATISFIED		
FTL-TC-02	Safety	G-(Safe1 && Safe2 && Safe3 && Safe4)	SATISFIED		
FTL-TC-03	Liveness	F(Green3    Green4)	SATISFIED		
FTL-TC-04	Liveness	F(Yellow2 && -Green2)	SATISFIED		
FTL-TC-05	Correlation	F(Red1) => F((Red1) && F(Red1 && Safe1))	SATISFIED		
FTL-TC-06	Correlation	F(Safe3 && Red3) => F((Safe3 && Red3) && X(Green3))	SATISFIED		
FTL-TC-07	Correlation	F(Safe2 && Safe3) => F((Safe2 && Safe3) && F(Safe4))	SATISFIED		
FTL-TC-08	Until	(Red1 && Safe1) U Green1	SATISFIED		
FTL-TC-09	Until	Red4 U Green4	SATISFIED		

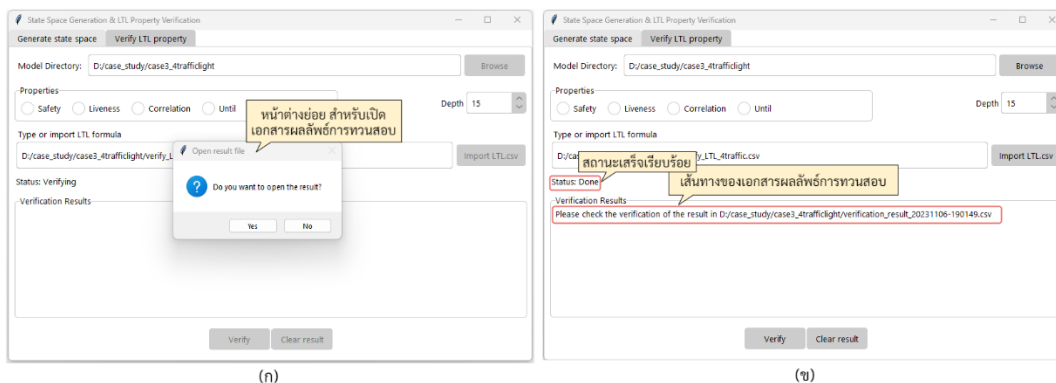
รูปที่ ข.13 ตัวอย่างรูปแบบของเอกสารสูตรแอลที่แอล

เมื่อเลือกเอกสารสูตรแอลที่แอลที่ต้องการทวนสอบแล้ว ไตรรกทอริของเอกสารสูตรแอลที่แอลแสดงในช่องข้อความ เมื่อผู้ใช้กดปุ่มการทวนสอบ “Verify” ระบบอยู่ในสถานะกำลังทวนสอบ คุณสมบัติที่อยู่ในเอกสารสูตรแอลที่แอลและปุ่มบนหน้าจอกการทวนสอบถูกปิดการใช้งาน แสดงหน้าจอกการดำเนินการทวนสอบเอกสารสูตรแอลที่แอลในรูปที่ ข.14



รูปที่ ข.14 หน้าจอแสดงระบบกำลังดำเนินการทวนสอบเอกสารสูตรแอลที่แอล

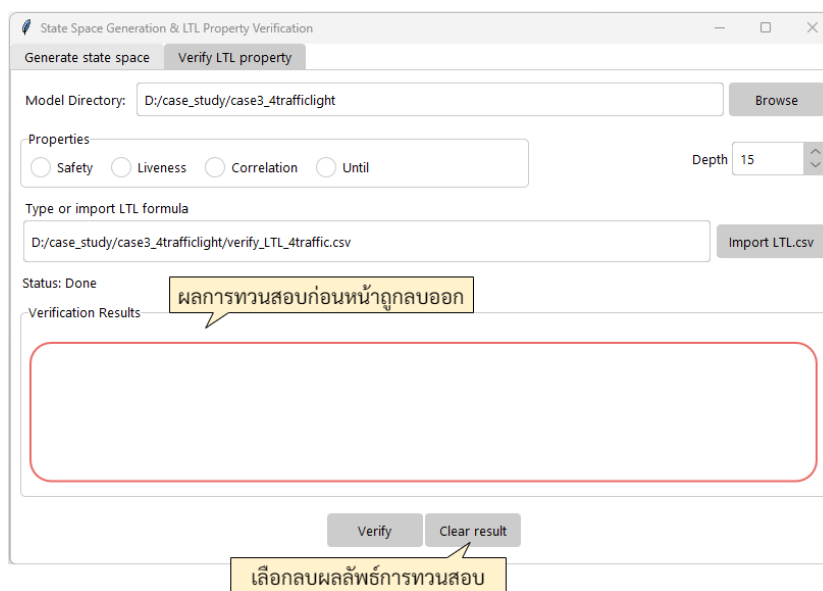
เมื่อระบบดำเนินการทวนสอบเสร็จเรียบร้อย ในรูปที่ ข.15 (ก) ระบบแสดงหน้าต่างย่อยสำหรับการเปิดเอกสารผลลัพธ์การทวนสอบโดยอัตโนมัติ หากผู้ใช้ไม่ต้องการเปิดทันที (ข) ระบบกลับเข้าสู่หน้าจอปกติและสถานะเสร็จเรียบร้อย พร้อมทั้งแสดงไตรรกทอริของเอกสารผลลัพธ์การทวนสอบที่ถูกบันทึก



รูปที่ ข.15 หน้าจอแสดงผลการทวนสอบเอกสารสูตรแอลที่แอล



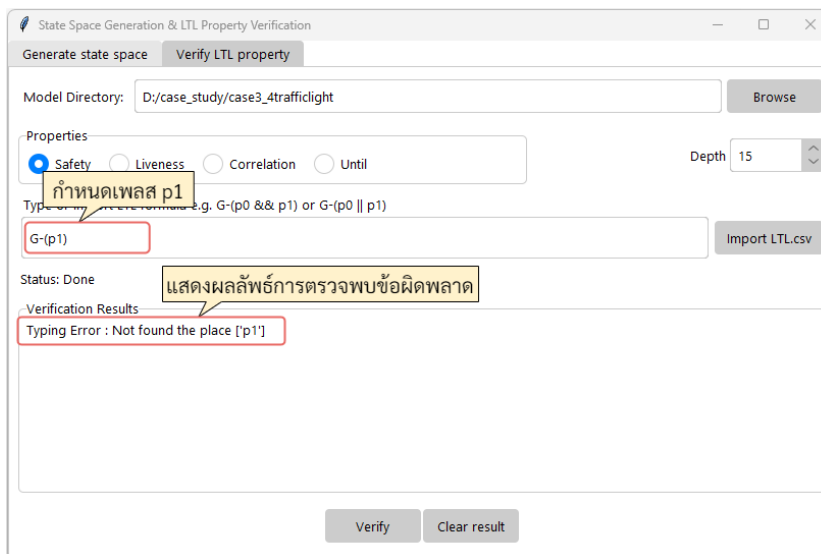
ในกรณีที่ต้องการลบผลการทวนสอบก่อนหน้าออก ผู้ใช้กดปุ่ม “Clear result” ระบบดำเนินการลบผลลัพธ์ก่อนหน้าออก แสดงในรูปที่ ข.16



รูปที่ ข.16 หน้าจอแสดงการลบผลการทวนสอบ

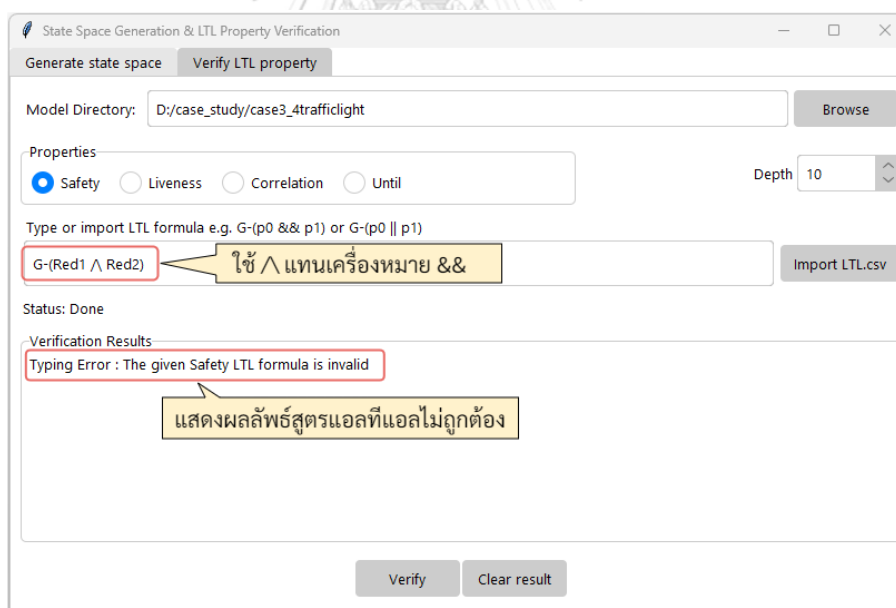
นอกจากนี้ ก่อนดำเนินการทวนสอบ สูตรแอลทีแอลที่ผู้ใช้กำหนดจะได้รับการตรวจสอบความถูกต้องก่อนดำเนินการทวนสอบว่าเป็นไปตามรูปแบบของแต่ละคุณสมบัติที่ต้องการทวนสอบหรือไม่ เช่น ชื่อเพลสที่ระบุในสูตรแอลทีแอลมีจริงในระบบหรือไม่ สูตรแอลทีแอลที่กำหนดสอดคล้องกับคุณสมบัติที่เลือกหรือไม่หรือตรวจสอบความสมมูลของวงเล็บ เป็นต้น

ในรูปที่ ข.17 ระบุชื่อเพลสที่ไม่มีอยู่จริงในระบบ เมื่อกดปุ่มการทวนสอบ ระบบตรวจพบข้อผิดพลาดและแสดงผลข้อผิดพลาดของชื่อเพลสที่ระบุในสูตรแอลทีแอล



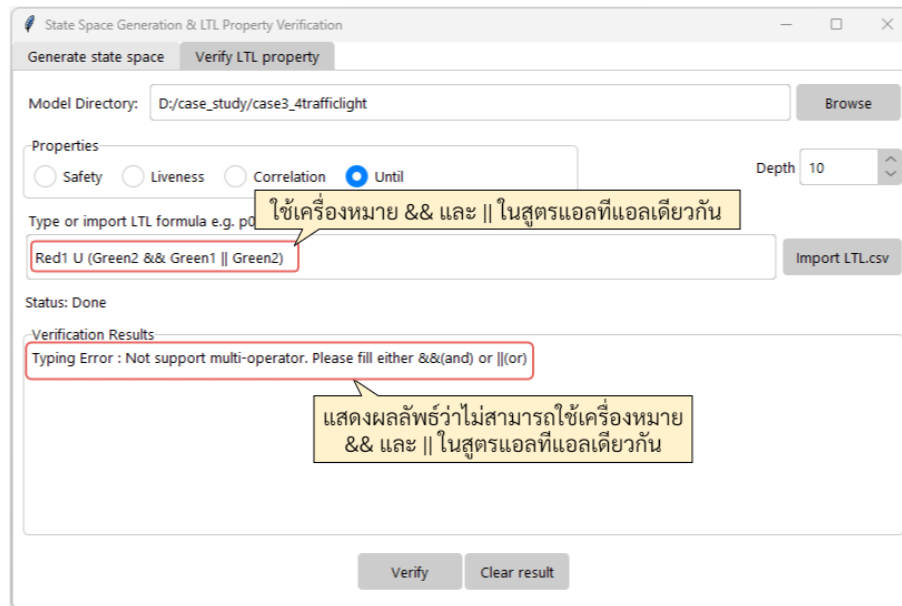
รูปที่ ข.17 หน้าจอแสดงผลการตรวจพบชื่อเพลสที่กำหนดไม่มีอยู่ในระบบ

ในรูปที่ ข.18 เครื่องหมายไม่ตรงตามที่กำหนด เมื่อกดปุ่มการทวนสอบ ระบบตรวจพบข้อผิดพลาดและแสดงผลสูตรแอลที่แอลที่กำหนดไม่ถูกต้อง



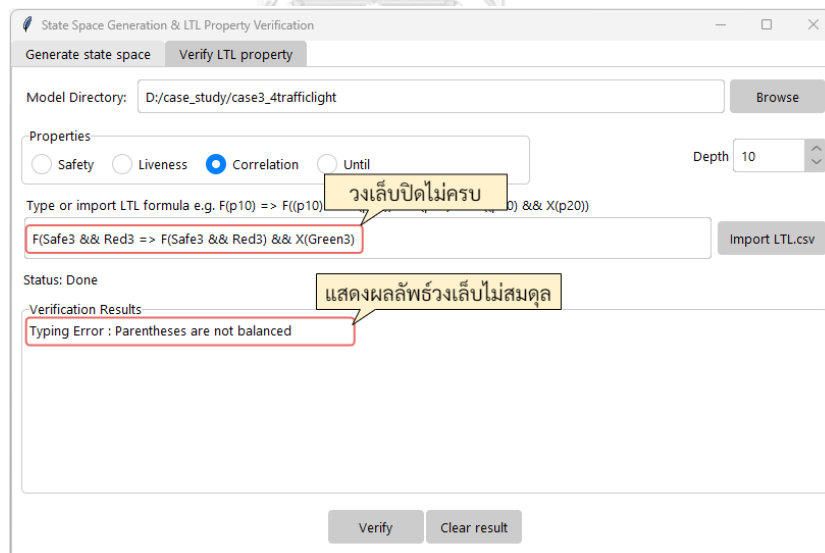
รูปที่ ข.18 หน้าจอแสดงข้อผิดพลาดเมื่อสูตรแอลที่แอลที่ระบุไม่ถูกต้อง

ในรูปที่ ข.19 สูตรแอลที่แอลใช้เครื่องหมาย && และ || พร้อมกัน เมื่อกดปุ่มการทวนสอบระบบตรวจพบข้อผิดพลาดและแสดงผลแจ้งเตือนไม่สามารถใช้เครื่องหมาย && และ || ในสูตรแอลที่แอลเดียวกัน



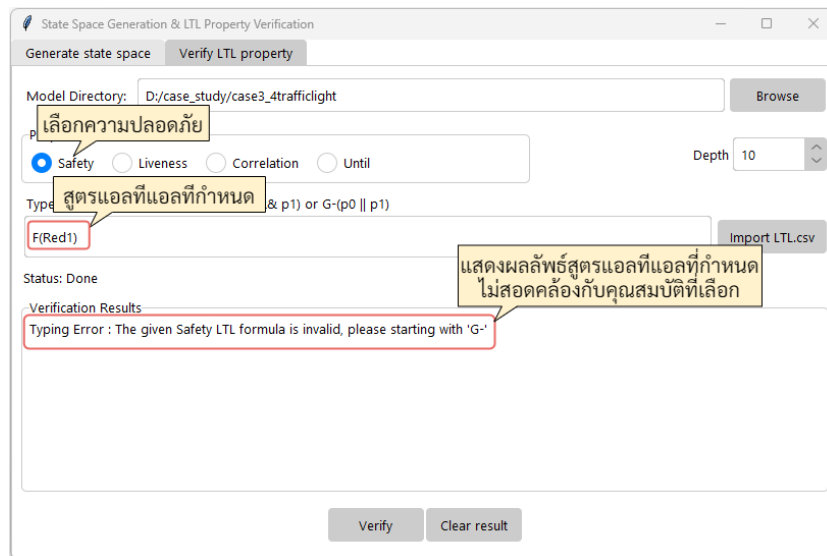
รูปที่ ข.19 หน้าจอแสดงข้อผิดพลาดเมื่อผู้ใช้ระบุเครื่องหมาย && และ || พร้อมกัน

ในรูปที่ ข.20 ระบุสูตรแอลทีแอลที่มีเครื่องหมายวงเล็บปิดสมดุล เมื่อกดปุ่มการทวนสอบ ระบบตรวจพบข้อผิดพลาดและแสดงผลวงเล็บไม่สมดุล



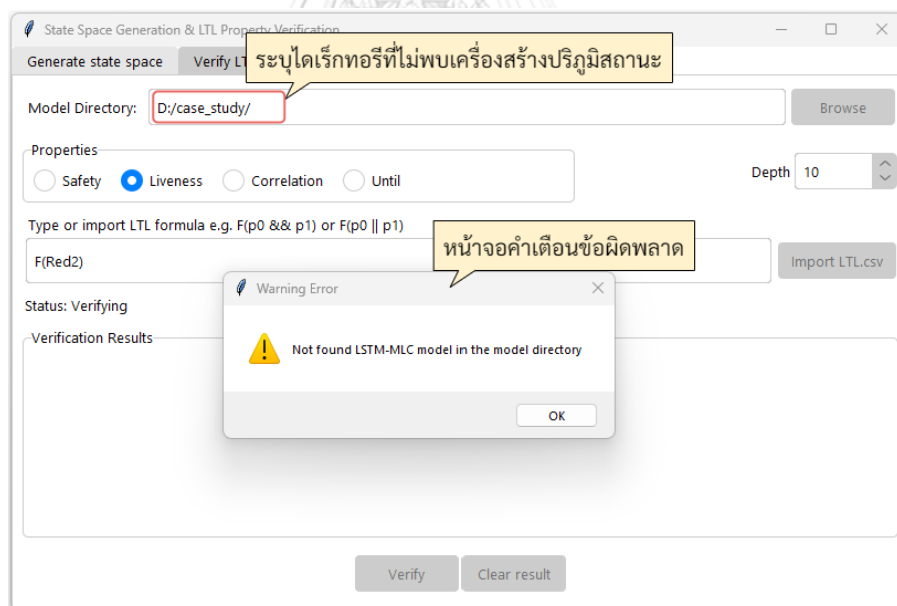
รูปที่ ข.20 หน้าจอแสดงผลข้อผิดพลาดเมื่อสูตรแอลทีแอลที่ระบุมีวงเล็บที่ไม่สมดุล

ในรูปที่ ข.21 ระบุสูตรแอลทีแอลที่ไม่สอดคล้องกับคุณสมบัติที่เลือก เมื่อกดปุ่มการทวนสอบ ระบบตรวจพบข้อผิดพลาดและแสดงผลสูตรแอลทีแอลไม่ถูกต้อง พร้อมแสดงคำแนะนำให้ใช้ตัวดำเนินการเชิงเวลาที่ยืนยันด้วย “G-”



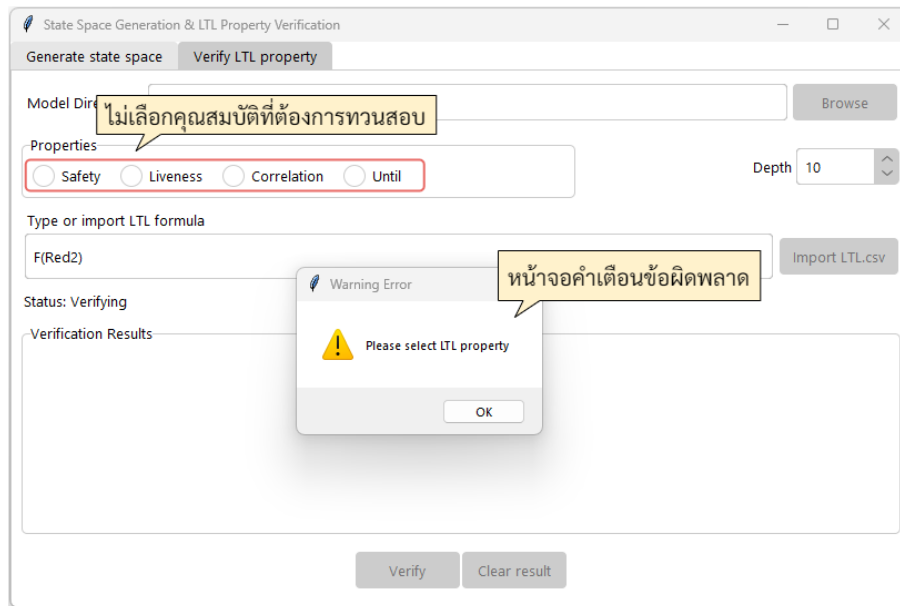
รูปที่ ข.21 หน้าจอแสดงผลข้อผิดพลาดเมื่อสูตรแอลทีแอลไม่สอดคล้องกับคุณสมบัติที่เลือก

ในรูปที่ ข.22 เมื่อใดเรกทอรีที่ผู้ใช้ระบุเข้ามาไม่พบเครื่องสร้างปริภูมิสถานะ ระบบแสดงหน้าต่างแจ้งเตือนข้อผิดพลาด



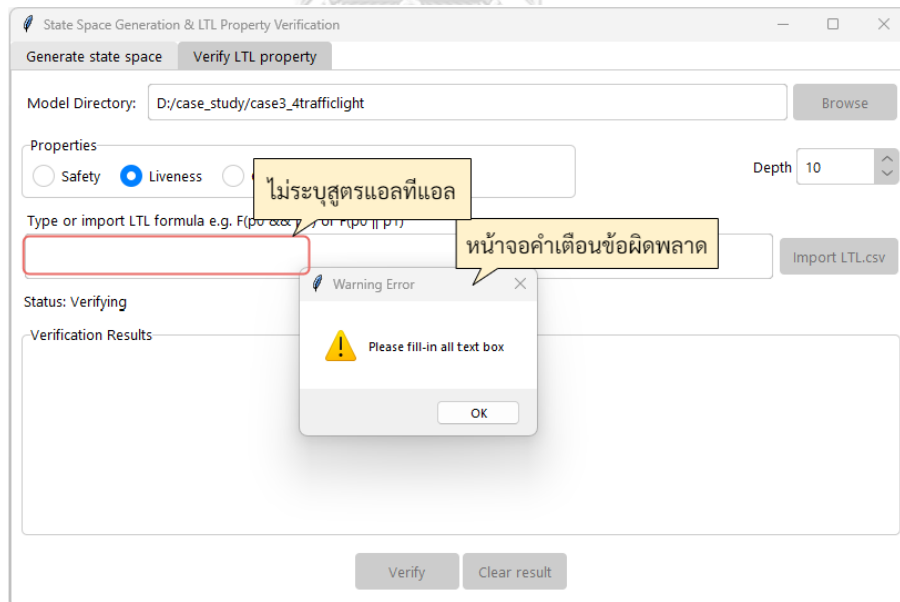
รูปที่ ข.22 หน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อไม่พบเครื่องสร้างปริภูมิสถานะจากไดเรกทอรีที่ระบุ

ในรูปที่ ข.23 เมื่อผู้ใช้ไม่ได้เลือกคุณสมบัติที่ต้องการทวนสอบ ระบบแสดงหน้าต่างแจ้งเตือนข้อผิดพลาดให้เลือกคุณสมบัติที่ต้องการทวนสอบ



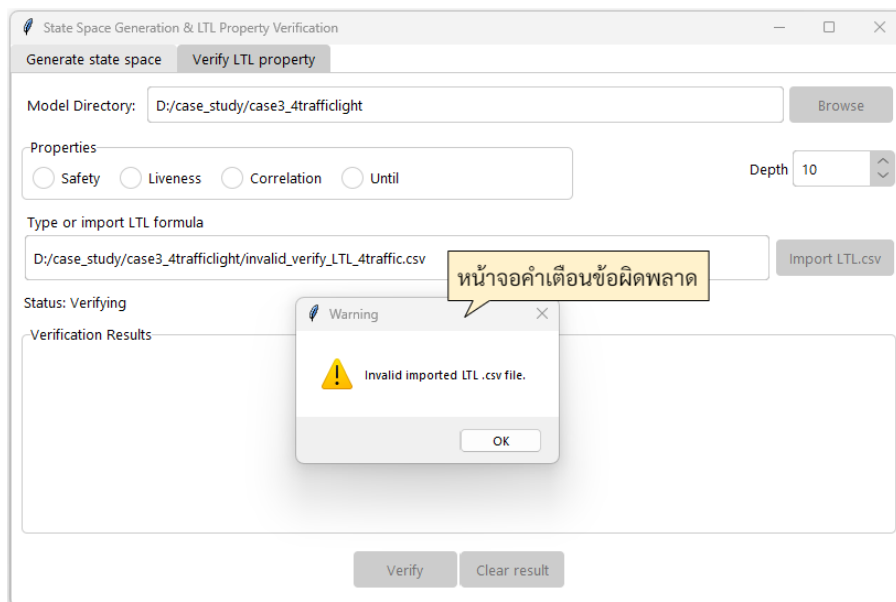
รูปที่ ข.23 หน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อผู้ใช้ไม่เลือกคุณสมบัติที่ต้องการทวนสอบ

ในรูปที่ ข.24 เมื่อผู้ใช้ไม่ระบุสูตรแอลทีแอลที่ต้องการทวนสอบ ระบบแสดงหน้าต่างแจ้งเตือนข้อผิดพลาดให้ระบุสูตรแอลทีแอลที่ต้องการทวนสอบ



รูปที่ ข.24 หน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อผู้ใช้ไม่ระบุสูตรแอลทีแอลที่ต้องการทวนสอบ

ในรูปที่ ข.25 เมื่อผู้เลือกนำเข้าเอกสารสูตรแอลทีแอล ระบบแสดงหน้าต่างแจ้งเตือนข้อผิดพลาดเอกสารสูตรแอลทีแอลไม่ถูกต้อง



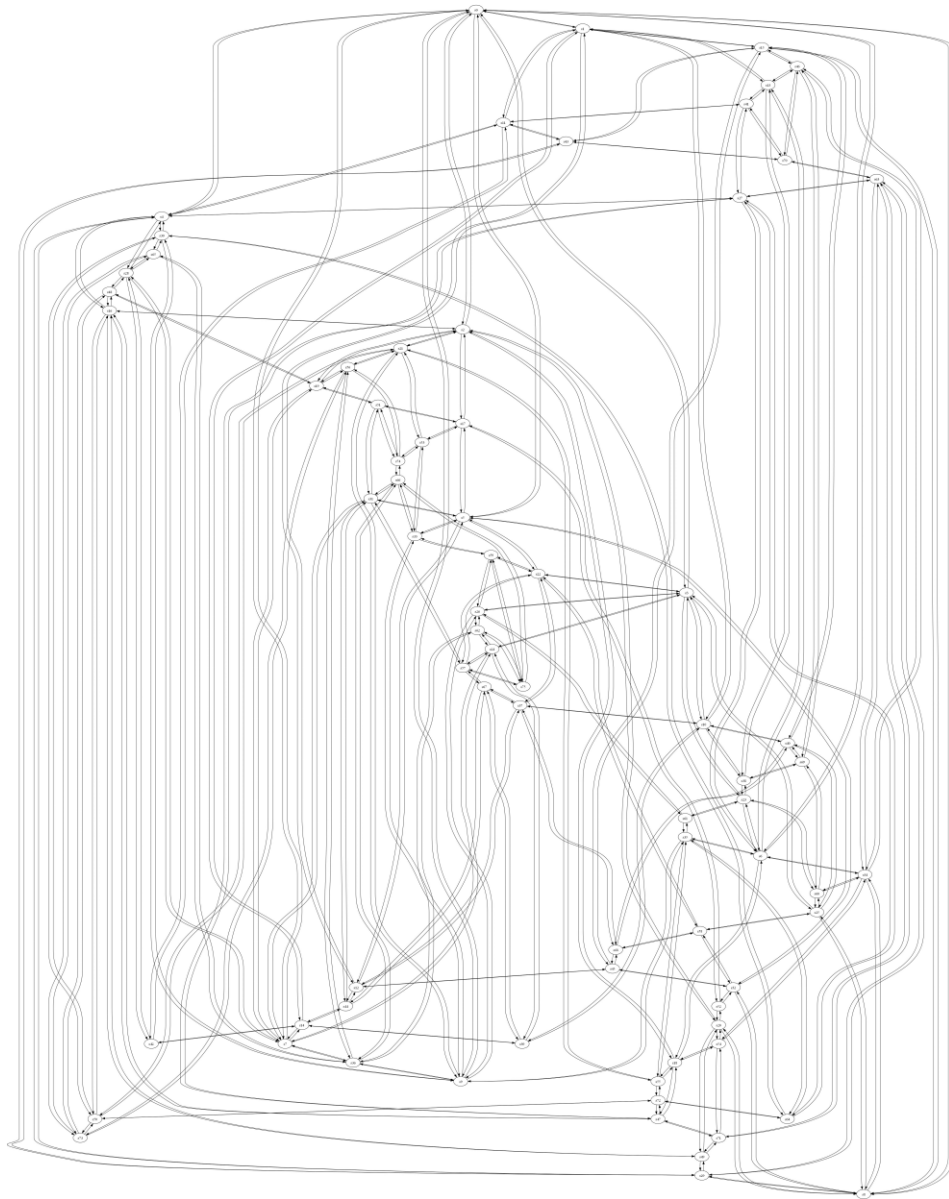
รูปที่ ข.25 หน้าต่างแจ้งเตือนข้อผิดพลาดเมื่อรูปแบบของเอกสารสูตรแอลทีแอลไม่ถูกต้อง



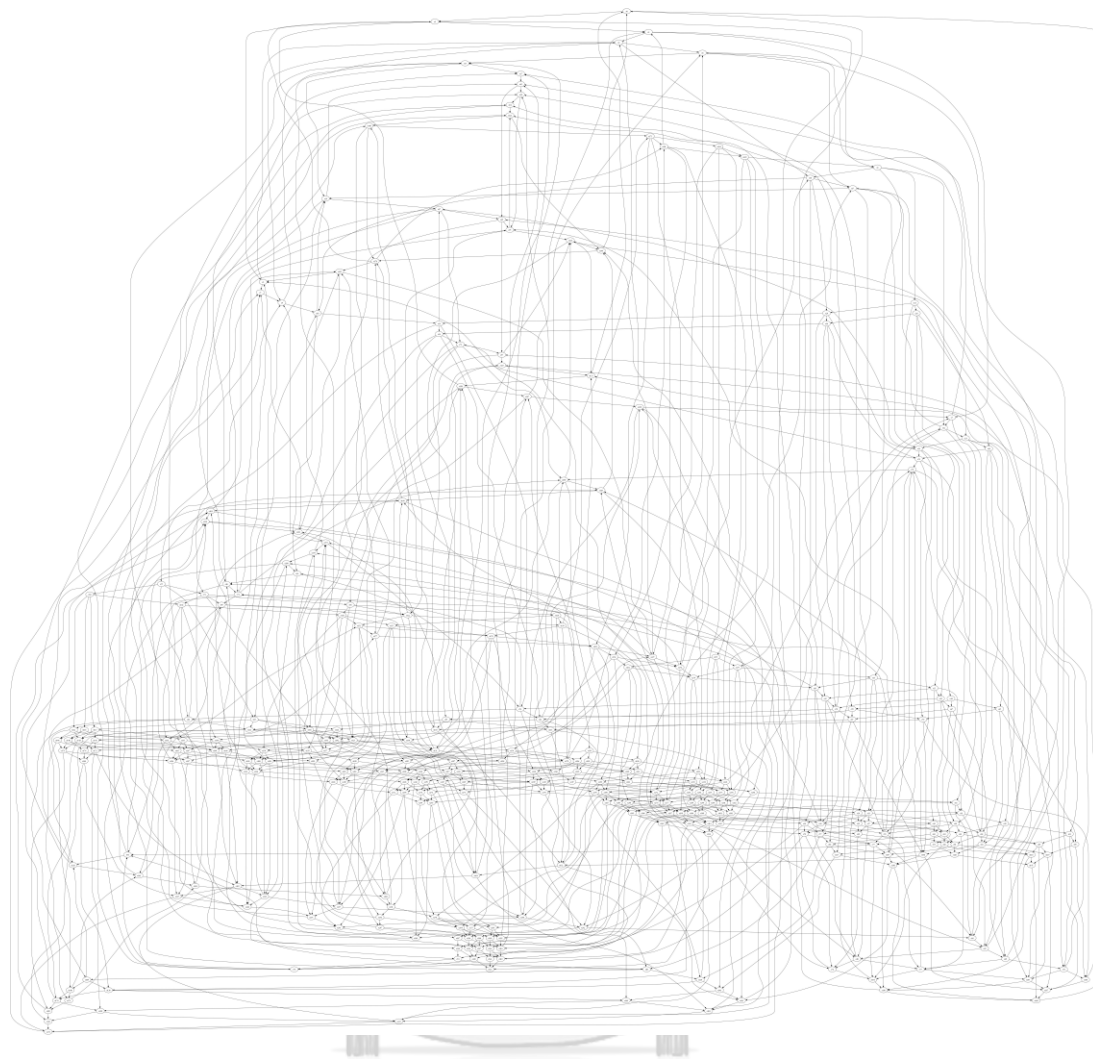
## ภาคผนวก ค.

## กราฟปริภูมิสถานะสำหรับกรณีศึกษาที่ 2 3 และ 4

สำหรับกราฟปริภูมิสถานะที่นำมาแสดง อาจจะไม่ชัดเจนเท่าไรนัก เนื่องจากจำนวนสถานะของกรณีศึกษามีจำนวนมาก โดยผู้วิจัยมีจุดประสงค์ที่นำมาแสดงเพื่อให้ผู้อ่านเห็นความซับซ้อนของกรณีศึกษาของงานวิจัยนี้ ซึ่งเป็นนการนำเสนอแผนภาพข้อมูล (Data visualization) ของกราฟปริภูมิสถานะสำหรับกรณีศึกษาที่ 2 3 และ 4 ดังต่อไปนี้



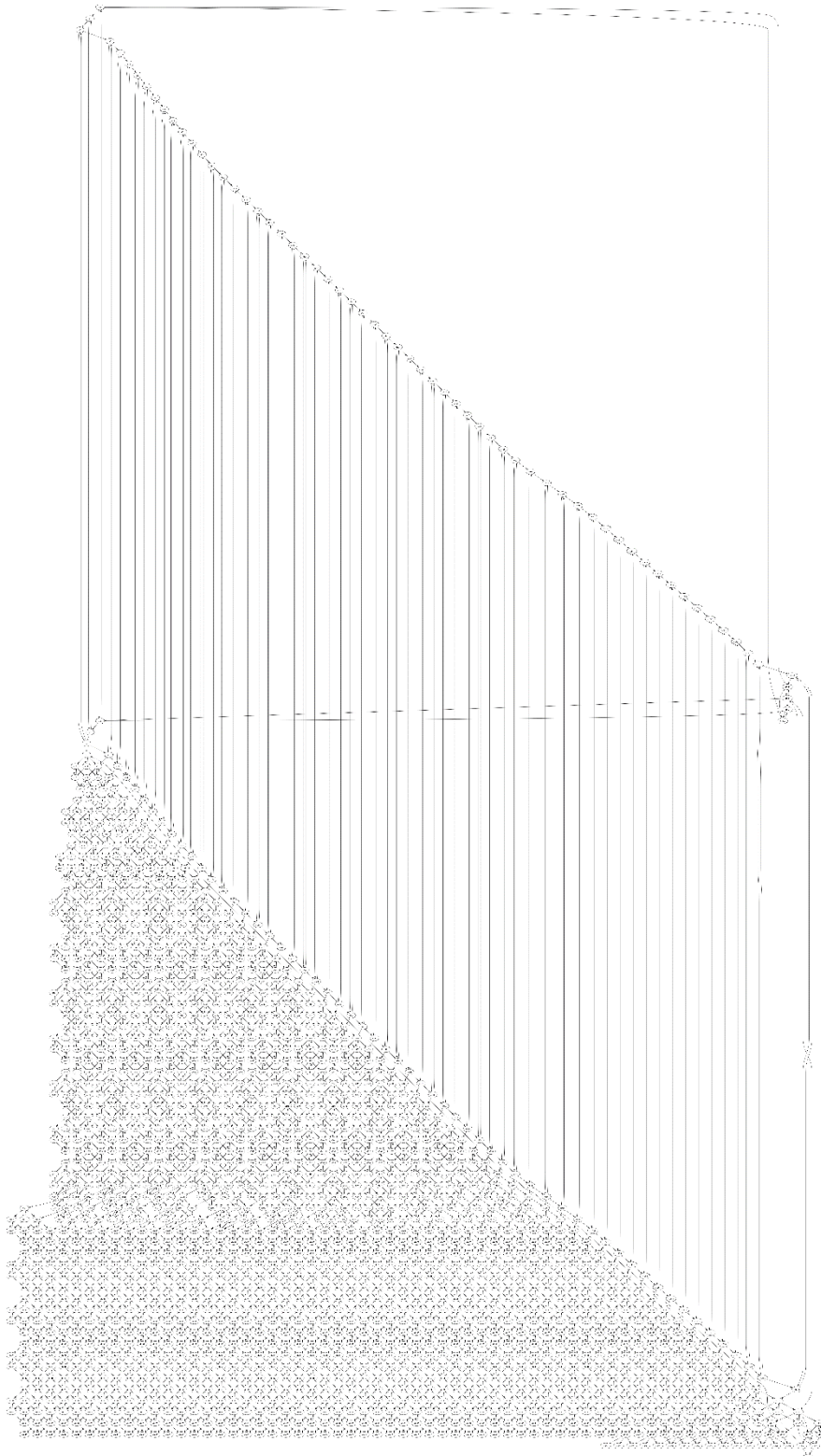
รูปที่ ค.1 การนำเสนอแผนภาพข้อมูลของกราฟปริภูมิสถานะของกรณีศึกษาที่ 2 ระบบการ  
รับประทานอาหารของนักปราชญ์ 9 คน



รูปที่ ค.2 การนำเสนอแผนภาพข้อมูลของกราฟปริภูมิสถานะของกรณีศึกษาที่ 3 ระบบสัญญาณไฟจราจรสี่ต้น

สำหรับการนำเสนอแผนภาพข้อมูลของกราฟปริภูมิสถานะของกรณีศึกษาที่ 4 ระบบหน่วยความจำดาวเทียม ที่แสดงใน รูปที่ ค.3 เป็นการแสดงของสถานะ  $s_0$  ถึง สถานะ  $s_{2000}$  เท่านั้น เนื่องจากกราฟปริภูมิสถานะของระบบนี้มีขนาดใหญ่เกินกว่าที่จะวาดออกมาเป็นแผนภาพข้อมูลได้ครบทุกสถานะ





รูปที่ ค.3 การนำเสนอแผนภาพข้อมูลของกราฟปริภูมิสถานะของกรณีศึกษาที่ 4 ระบบหน่วยความจำดาวเทียม

## ภาคผนวก ง.

## ผลประเมินประสิทธิภาพการตรวจสอบความถูกต้องของข้อมูลนำเข้าโดยใช้เครื่องมือการทวนสอบ

จากที่ได้ประเมินประสิทธิภาพของการทำงานของเครื่องมือการทวนสอบโดยการนำผลการทวนสอบไปเปรียบเทียบกับผลการทวนสอบที่คาดหวังแล้ว ผู้วิจัยได้ประเมินความสามารถของเครื่องมือการทวนสอบสำหรับการตรวจสอบข้อมูลนำเข้าสู่ตราแอลทีแอลจากผู้ใช้ ซึ่งแสดงผลการประเมินในตารางที่ ง.1 จะเห็นได้ว่าเครื่องมือการทวนสอบสามารถตรวจจับข้อผิดพลาดข้อมูลนำเข้าได้อย่างถูกต้อง

ตารางที่ ง.1 ผลประเมินประสิทธิภาพการตรวจสอบข้อมูลนำเข้าโดยใช้เครื่องมือการทวนสอบ

รหัสทดสอบ	คุณสมบัติ	สูตรแอลทีแอล	ผลการตรวจสอบข้อมูลนำเข้า
VAL-01	ความปลอดภัย	$A(p_0 \parallel p_1)$	ตรวจพบข้อผิดพลาด
VAL-02	ความปลอดภัย	$G(p_0 \parallel p_1)$	ตรวจพบข้อผิดพลาด
VAL-03	ความปลอดภัย	$G-(p_0 \wedge p_1)$	ตรวจพบข้อผิดพลาด
VAL-04	ความปลอดภัย	$G-(p_0 \&\& p_1 \parallel)$	ตรวจพบข้อผิดพลาด
VAL-05	ความปลอดภัย	$G-(p_0 \&\& p_1 \parallel p_2)$	ตรวจพบข้อผิดพลาด
VAL-06	ความปลอดภัย	$G-(p_{100})$	ตรวจพบข้อผิดพลาด
VAL-07	ความปลอดภัย	$G-(p_1 p)$	ตรวจพบข้อผิดพลาด
VAL-08	ความดำเนินชีวิต	$G(p_0)$	ตรวจพบข้อผิดพลาด
VAL-09	ความดำเนินชีวิต	$(p_0 \parallel p_1)$	ตรวจพบข้อผิดพลาด
VAL-10	ความดำเนินชีวิต	$F(p_0 \wedge p_1)$	ตรวจพบข้อผิดพลาด
VAL-11	ความดำเนินชีวิต	$F(p_0 \&\& p_1 \parallel)$	ตรวจพบข้อผิดพลาด
VAL-12	ความดำเนินชีวิต	$F(p_0 \&\& p_1 \parallel p_2)$	ตรวจพบข้อผิดพลาด
VAL-13	ความดำเนินชีวิต	$F(p_0))$	ตรวจพบข้อผิดพลาด
VAL-14	ความสัมพันธ์	$((p_0) \Rightarrow F((p_0) \&\& F(p_2)))$	ตรวจพบข้อผิดพลาด
VAL-15	ความสัมพันธ์	$F(p_0) \Rightarrow a((p_0) \&\& F(p_2))$	ตรวจพบข้อผิดพลาด
VAL-16	ความสัมพันธ์	$F(p_0) \Rightarrow F((p_0) \&\& z(p_2))$	ตรวจพบข้อผิดพลาด
VAL-17	ความสัมพันธ์	$F() \Rightarrow F((p_0) \&\& F(p_2))$	ตรวจพบข้อผิดพลาด
VAL-18	ความสัมพันธ์	$F(p_0) \Rightarrow F(() \&\& F(p_2))$	ตรวจพบข้อผิดพลาด
VAL-19	ความสัมพันธ์	$F(p_0) \Rightarrow F((p_0) \&\& F())$	ตรวจพบข้อผิดพลาด

ตารางที่ ง.1 ผลประเมินประสิทธิภาพการตรวจสอบข้อมูลนำเข้าโดยใช้เครื่องมือการทวนสอบ (ต่อ)

รหัสทดสอบ	คุณสมบัติ	สูตรแอลทีแอล	ผลการตรวจสอบข้อมูลนำเข้า
VAL-20	ความสัมพันธ์	$F(p0) \Rightarrow F((p0) \parallel F(p2))$	ตรวจพบข้อผิดพลาด
VAL-21	ความสัมพันธ์	$F(p0) \Rightarrow F((p0) \&\& F(p2 \text{ or } p1))$	ตรวจพบข้อผิดพลาด
VAL-22	ความสัมพันธ์	$F(p0) \Rightarrow F((p1) \&\& F(p2))$	ตรวจพบข้อผิดพลาด
VAL-23	ความสัมพันธ์	$F(p0) \rightarrow F((p0) \&\& F(p2))$	ตรวจพบข้อผิดพลาด
VAL-24	ประเภท Until	$p0$	ตรวจพบข้อผิดพลาด
VAL-25	ประเภท Until	$p0 \cup p1$	ตรวจพบข้อผิดพลาด
VAL-26	ประเภท Until	$p0 \cup p1 \wedge p2$	ตรวจพบข้อผิดพลาด
VAL-27	ประเภท Until	$p0 \cup p1 \&\& p2 \parallel p3$	ตรวจพบข้อผิดพลาด
VAL-28	ประเภท Until	$p0 \vee p1 \cup p2$	ตรวจพบข้อผิดพลาด
VAL-29	ประเภท Until	$p0 \cup (p1$	ตรวจพบข้อผิดพลาด
VAL-30	ประเภท Until	$(p0 \cup (p1$	ตรวจพบข้อผิดพลาด

## ภาคผนวก จ.

## อภิธานศัพท์

กรณีศึกษา	Case study
กระบวนการภาษาธรรมชาติ	Natural language processing
กระบวนการทวนสอบเชิงรูปนัย	Formal verification process
กระบวนการโทเคนไนเซชัน	Tokenization process
กระบวนการหลัก	Core process
กราฟการเข้าถึง	Reachability graph
กราฟแบบมีทิศทาง	Directed graph
กลไกความสนใจ	Attention mechanism
กลยุทธ์การจัดเก็บ	Storing strategy
กลยุทธ์การหาขีดแบ่ง	Threshold strategy
ก้อน	Chunk
การเข้าถึงมาร์กกิง	Reachable marking
การเข้าถึงมาร์กกิงทั้งหมด	All reachability marking
การค้นหาค้นหาเป็นแนวกว้าง	Breadth first search
การค้นหาค้นหาเป็นแนวลึก	Depth first search
การจัดหมวดหมู่ข้อความ	Text categorization
การจำแนกแบบหลายขั้นตอน	Multi-step multi-classification
การจำแนกแบบหลายคลาส	Multi-class classification
การจำแนกแบบหลายป้าย	Multi-label classification
การจำลองที่ละขั้นตอน	Simulation
การตรวจสอบความถูกต้องข้อมูลนำเข้า	Input validation
การตรวจสอบความสมมูล	Equivalence checking
การตัดแต่งกิ่งปริภูมิสถานะ	State space pruning
การถดถอยโลจิสติกส์	Logistic regression
การทวนสอบ	Verification

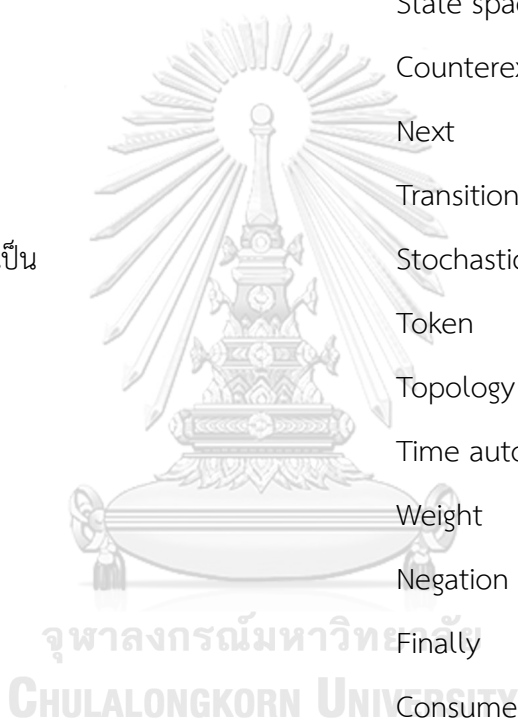
การทวนสอบเชิงรูปนัย	Formal verification
การทวนสอบอย่างไม่เป็นทางการ	Informal verification
การทำความสะอาดข้อมูล	Data cleaning
การนำเสนอแผนภาพข้อมูล	Data visualization
การแปลงข้อมูล	Data transformation
การพิสูจน์ทฤษฎีบท	Theorem proving
การเพิ่มประสิทธิภาพตามการค้นหา	Search-based optimization
การระเบิดของปริภูมิสถานะ	State space explosion
การรู้จำคำพูด	Speech recognition
การเรียนรู้แบบเสริมกำลัง	Reinforcement learning
การเรียนรู้ไวยากรณ์	Grammar learning
การลดความสมมาตร	Symmetry reduction
การลดบางส่วน	Partial reduction
การลดสถานะ	State reduction
การละเมิดการยืนยัน	Assertion violations
การเลือกข้อมูล	Data Selection
การวินิจฉัยโรคทางการแพทย์	Diagnosis disease for medical
การหยุดชะงัก	Deadlocks
การหารวม	Union
การหาส่วนร่วม	Intersection
การให้เหตุผล	Implication
กำลังทวนสอบ	Verifying
กำลังสร้าง	Generating
ข้อมูลไม่สมดุล	Imbalance data
ข้อมูลอนุกรมเวลา	Time series data
เข้ารหัสแบบวันฮอต	One-hot encoding
ความจำเพาะ	Specificity



ความดำเนินชีวิต	Liveness
ความต้องการเชิงฟังก์ชัน	Functional requirement
ความต้องการที่ไม่เป็นฟังก์ชัน	Non-Functional requirement
ความปลอดภัย	Safety
ความผิดพลาดของมนุษย์	Human error
ความไว	Sensitivity
ความสัมพันธ์	Correlation
ความสามารถในการใช้งาน	Usability
ค่าความเที่ยงระดับไมโคร	Micro-Precision
ค่าระยะที่หายไป	Hamming Loss
ค่าเรียกคืนระดับไมโคร	Micro-Recall
ค่าสูงที่สุด	Maximum Cut
ค่าอคติ	Bias
คุณลักษณะอินพุต	Input feature
คุณสมบัติความเป็นธรรม	Fairness property
คุณสมบัติความสมบูรณ์	Sound property
คุณสมบัติชั่วคราว	Temporal properties
คุณสมบัติด้านความดำเนินชีวิต	Liveness property
คุณสมบัติด้านความปลอดภัย	Safety property
คุณสมบัติด้านความสัมพันธ์	Correlation property
เครื่องมือซีพีเอ็น	CPN tool
เครื่องมือทีน่า	TINA tool
เครื่องมือเพิ่มประสิทธิภาพพอดัม	Adam optimizers
เครื่องมือมินิคอนด้า 3	Miniconda3
เครื่องมือสปิน	Spin tool
เครื่องสร้างปริภูมิสถานะ	State space generator
โครงข่ายความจำระยะสั้นแบบยาว	Long short-term memory network

โครงข่ายประสาทเทียม	Artificial neural network
โครงข่ายประสาทเทียมแบบกราฟ	Graph neural network
โครงข่ายประสาทเทียมแบบคอนโวลูชัน	Convolutional neural network
โครงข่ายประสาทเทียมแบบวนซ้ำ	Recurrent neural network
โครงสร้างคริปกี	Kripke structure
จนกระทั่ง	Until
จินเนติกอัลกอริทึม	Genetic algorithms
จุดตัด	Cut-off point
จุดยอด	Vertex
เจ-อินเด็กซ์	J-index
ชั้น	Layer
ชั้นการจำแนกแบบหลายป้าย	Multi-label prediction layer
ชั้นการฝัง	Embedding layer
ชั้นผลลัพธ์	Output layer
ชั้นแอลเอสทีเอ็ม	LSTM layer
ชุดของเส้นทาง	Flow relation
ชุดข้อมูล	Dataset
ชุดข้อมูลการตรวจสอบ	Validation dataset
ชุดข้อมูลการทดสอบ	Test dataset
ชุดข้อมูลการฝึกอบรม	Training dataset
ชุดข้อมูลดิบ	Raw dataset
เซกเตอร์ที่ใช้ได้	Available Sector
เซตของสถานะที่สอดคล้องกัน	Set of corresponding states
ต้นไม้การตัดสินใจ	Decision tree
ต้นไม้เชิงการคำนวณ	Computational tree
ตรรกศาสตร์เชิงเวลาแบบลำดับ	Linear temporal logic
ตรรกศาสตร์ต้นไม้เชิงการคำนวณ	Computational tree logic

ตัวคูณ	Element-wise multiplication
ตัวจัดการปริภูมิสถานะ	State space manager
ตัวจัดการคุณสมบัติแอลทีแอล	LTL property manager
ตัวยืนยัน	Invariant
ตัววัดการประเมินผลของการจำแนกหลายป้าย	Evaluation metric for multi-label classification
ตัวแสดงคุณสมบัติแอลทีแอล	LTL property visualizer
ตัวแสดงปริภูมิสถานะ	State space visualizer
ตัวอย่างค้าน	Counterexample
ถัดไป	Next
ทรานซิชัน	Transition
เทคนิคสุ่มความน่าจะเป็น	Stochastic technique
โทเคน	Token
โทโพโลยีของกราฟ	Topology of graph
ไทม์ออโตมาตา	Time automata
น้ำหนัก	Weight
นิเสธ	Negation
ในที่สุด	Finally
บริโภค	Consume
บัฟเฟอร์วงกลม	Circular buffer
บี แมชชีน	B machine
บูตทรีส์	Boosted tree
แบบจำลองความสัมพันธ์ไบนารี	Binary relevance model
แบบจำลองความสัมพันธ์ไบนารีแบบเพื่อนบ้านที่ใกล้ที่สุดแบบหลายป้าย	Binary relevance of K-nearest Neighbors model
แบบจำลองเชิงรูปนัยของระบบ	Formal system model
แบบจำลองพื้นฐาน	Baseline model
แบบจำลองเพื่อนบ้านที่ใกล้ที่สุดแบบหลายป้าย	Multi-label K-nearest neighbor





แบบจำลองแอลเอสทีเอ็ม-เอ็มแอลซี	model
ไบนารีครอสเอนโทรปี	LSTM-MLC model
ไบนารีเวกเตอร์	Binary cross entropy
ประตู	Binary vector
ประตูทางเข้า	Gate
ประตูทางออก	Input gate
ประตูลืม	Output gate
ประพจน์เดี่ยว	Forget gate
ปริภูมิสถานะ	Atomic proposition
ปัญหาการสูญหายของเกรเดียนต์	State space
ป้ายกำกับเอาต์พุต	Vanishing gradient problem
ผลบวกจริง	Output label
ผลบวกปลอม	True positive
ผลลบจริง	False positive
ผลลบปลอม	True negative
ผลลัพธ์ที่คาดหวัง	False negative
ผลลัพธ์ที่แท้จริง	Expected output
ผลิต	Actual output
ผิดพลาด	Produce
ผู้ใช้งาน	Error
ผู้ทดสอบ	User
แผนภาพดีพลอยเมนต์	Tester
แผนภาพยูสเคส	Deployment diagram
แผนภาพระบบย่อย	Use case diagram
พจนานุกรมโทเคน	Subsystem diagram
พร้อมใช้	Tokenization dictionary
	Ready

พอใจ	Satisfied
พอยน์เตอร์การเขียน	Writing pointer
พอยน์เตอร์การอ่าน	Reading pointer
พื้นที่ใต้เส้นโค้งบนกราฟ	Receiver operating characteristic
เพทรีเน็ต	Petri net
เพทรีเน็ตแบบธรรมดา	Ordinary Petri net
เพลส	Place
โพรเมล่า	Promela
ไพธอน	Python
ฟังก์ชันการแมปเพลสไปยังสถานะ	Place-to-State mapping function
ฟังก์ชันซอฟต์แวร์แมกซ์	Softmax function
ฟังก์ชันซิกมอยด์	Sigmoid function
ฟังก์ชันอีแอลเอฟ	ELF function
ฟี หรือ ฟายล์	Phi
ไฟล์การกำหนดค่า	Configuration file
ภาระ	Overhead
มองไปข้างหน้า	Lookahead
มองเห็น	Visible
มาร์กกิงเริ่มต้น	Initial marking
โมเดลเช็กกิง	Model checking
ไม่พอใจ	Not satisfied
ยูเดน เจ อินเด็กซ์	Youden's J Index
รหัสโทเคน	Token ID
ระบบกระจาย	Distributed system
ระบบการรับประทานอาหารของนักปราชญ์ 9 คน	Dinner 9 philosopher system
ระบบเครื่องรับจ่ายเงินอัตโนมัติ	Automated teller machine system
ระบบที่สำคัญ	Critical systems

ระบบแบบจำกัด	Finite system
ระบบแบบไม่จำกัด	Infinite system
ระบบพร้อมกัน	Concurrent system
ระบบเรียลไทม์	Real time system
ระบบสัญญาณไฟจราจรสี่ต้น	Four traffics lights system
ระบบหน่วยความจำดาวเทียมแบบ	Satellite memory system
แรนดอมฟอเรส	Random forest
ลำดับต่อลำดับ	Sequence to sequence
และ	Conjunction
วัตถุ	Object
วัตถุและการกำหนดค่าเสริม	Additional object and configuration
วิทยาการคอมพิวเตอร์	Computer science
วิธีการจัดลำดับ	Rank-based method
วิธีการเชิงรูปนัย	Formal method
วิธีการบูสตี้ง	Boosting method
วิธีการรวมกลุ่ม	Ensemble methods
วิธีการสำรวจ	Traverse method
วิธีการหน้าต่างบานเลื่อน	Sliding window method
วิธีใช้คะแนน	Score-based method
วิธีทางสถิติแบบเกาส์เซียน	Gaussian statistical approaches
วิศวกรรมควบคุม	Control engineering
เวกเตอร์ทวิภาค	Binary vector
เศรษฐมิติ	Econometrics
สถานะ	State
สถานะการยิง	Firing state
สถานะจำกัด	Finite state
สถานะเปิดใช้งาน	Enable state

สถานะสุดท้าย	Final states
สิ่งประดิษฐ์	Artifact
สูตรแอลที่แอลที่เหมาะสม	Well-formed LTL
สูตรแอลที่แอลแบบเฉพาะ	Specific LTL formula
เส้นเชื่อม	Arcs
เส้นทางดำเนินงานที่ไม่สิ้นสุด	Infinite execution paths
เส้นทางย่อย	Sub-path
เสมอ	Always
เสร็จเรียบร้อย	Done
หมดจด	Exhaustive
หรือ	Disjunction
โหนดใบ	Leaf node
โหนดเริ่มต้น	Root node
ออน-เดอะ-ฟลาย	On-the-fly
อัตราการเรียนรู้	Learning rate
อัตราบวกจริง	True positive rate
อัตราบวกที่เท็จ	False positive rate
อันเดอร์ฟิตติ้ง	Underfitting
อัลกอริทึมแบบอาณานิคมมด	Ant colony optimization
อัลกอริทึมเพื่อนบ้านใกล้ที่สุด	K-nearest neighbor algorithm
อินพุตเพลส	Input place
เอกภพสัมพัทธ์	Universal set
เอพีไอและไลบรารีเสริม	Additional API and library
เอฟวันระดับไมโคร	Micro-F1
เอาต์พุตเพลส	Output place
โอเวอร์ฟิตติ้ง	Overfitting
ไฮเพอร์โบลิกแทนเจนต์	Hyperbolic tangent

## ประวัติผู้เขียน

ชื่อ-สกุล	ชลิกา ศักดิ์สุภาวัฒนกุล
วัน เดือน ปี เกิด	4 ตุลาคม 2534
สถานที่เกิด	ขอนแก่น
วุฒิการศึกษา	- วิทยาศาสตรบัณฑิต (วท.บ.) สาขาวิชาวิทยาการคอมพิวเตอร์ คณะ วิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ เมื่อปีการศึกษา 2557 - วิทยาศาสตรมหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ (วท.ม.) คณะ วิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2561 - และเข้าศึกษาในหลักสูตร วิศวกรรมศาสตรดุษฎีบัณฑิต (วศ.ด.) สาขา วิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปี การศึกษา 2562
ที่อยู่ปัจจุบัน	เลขที่ 46/399 ถ.อโศก-ดินแดง แขวงบางกะปิ เขตห้วยขวาง กรุงเทพฯ 10310
รางวัลที่ได้รับ	ทุนสนับสนุนการศึกษาหลักสูตรดุษฎีบัณฑิต สำหรับนิสิตระดับปริญญาเอก ศักยภาพสูง จาก กองทุนโครงการสร้างเสริมพลังจุฬาฯ ก้าวสู่ศตวรรษที่ 2 (Second Century Fund, C2F)