บทนำ

I. ความสำคัญของทฤษฎีการคำนวณ

ทฤษฎีการคำนวณเป็นศาสตร์ทางด้านวิทยาศาสตร์ที่ตระหนักถึงการเรียนรู้ คุณสมบัติทั่วๆ ไปของการคำนวณทั้งที่เป็นรูปแบบตามธรรมชาติ ทำโดยมนุษย์ หรือแม้กระทั่งการจินตนาการ โดยพยายามมุ่งมั่นให้ผู้เรียนมีความเข้าใจในธรรมชาติของการคำนวณที่มีประสิทธิภาพ นอกจากนี้ยังช่วยให้รู้จักฝึกหาวิธีที่ดีที่สุดในการแก้ปัญหาที่ต้องการด้วย

สำหรับทฤษฎีการคำนวนนี้ได้มีส่วนเข้าไปเกี่ยวข้องในหลายสาขาวิชา ที่เห็นได้ชัดเช่น ทฤษฎีความซับซ้อน (Complexity Theory) ซึ่งจะเกี่ยวข้องกับการพิจารณาทรัพยากรในการคำนวณโดยมองความซับซ้อนของงาน และอัลกอริทึม (Algorithm) ซึ่งมองหาวิธีแก้ไขปัญหาที่เผชิญอยู่

สำหรับทฤษฎีความซับซ้อนยังแบ่งย่อยเป็น ตัวแบบและทรัพยากรทางด้านการคำนวณเช่น เวลา พื้นที่ ความซับซ้อนของวงจร (Circuit) ส่วนอัลกอริทึมก็สามารถแบ่งย่อยเป็นงานต่างๆ ได้ เช่น

อัลกอริทึมของกราฟ (Graph Algorithm) การโปรแกรมเชิงเส้น (Linear programming) อัลกอริทึมการประมาณค่า (Approximation Algorithm) ทฤษฎีการคำนวณตัวเลข (Computational number theory) เรขาคณิตการคำนวณ (Computational geometry) เป็นต้น

จุดมุ่งหมายสูงสุดสำหรับทฤษฎีการคำนวณเป็นไปตามคำถามที่ว่า "What are the fundamental capabilities and limitations of a computer?" ซึ่งจะได้ศึกษาในเนื้อหาที่เกี่ยวข้องกับทฤษฎีทางศาสตร์คอมพิวเตอร์ ดังนี้

1. ทฤษฎีออโตมาตา (Automata Theory) เป็นทฤษฎีที่ตรวจสอบเพื่อให้ได้มาซึ่งคำตอบของคำถามที่ว่า "What is a computer? และ What can it do?" โดยจะศึกษาในเรื่องของ ออโตมาตาจำกัด (Finite Automata) ออโตมาตากดลง (Pushdown Automata) และเครื่องทัวริง (Turing Machine) รวมทั้งความสามารถและข้อจำกัดของเครื่องแต่ละแบบ

2. ทฤษฎีความสามารถในการคำนวณ (Computability Theory) เป็นทฤษฎีที่ตรวจสอบเพื่อให้ได้มาซึ่งคำตอบของคำถามที่ว่า "What is the most powerful model of computation? และ What can it do?"

3. ทฤษฎีความซับซ้อน (Complexity Theory) เป็นทฤษฎีที่พิจารณาตามคำถามที่ว่า "How long does it take to solve a particular problem" ซึ่งประกอบไปด้วยปัญหาแบบ P และ NP

จะเห็นว่าทฤษฎีการคำนวณประกอบด้วยทฤษฎีมากมายโดยมากจะมองไม่เห็นเด่นชัดถึงประโยชน์เชิงปฏิบัติในการแก้ปัญหา ดังนั้นผู้เขียนจึงเห็นว่ามันมีความจำเป็นที่จะต้องโน้มน้าวให้ผู้อ่านหรือผู้ที่ศึกษาในสาขาวิทยาการคอมพิวเตอร์ได้เข้าใจในเหตุผลว่า "ทำไมจึงต้องศึกษาวิชาทฤษฎีการคำนวณ"

เหตุผลหนึ่งที่ควรศึกษาทฤษฎีคือ ทฤษฎีทำให้เห็นภาพรวมของวิชา ในวิชาทางวิทยาการคอมพิวเตอร์นั้นมีรายละเอียดที่สำคัญมากมายที่ควรจะต้องจำและถ้าไม่มีวิธีการรวบรวมรายละเอียดเหล่านี้ก็จะเป็นการยากที่จะจดจำ ทฤษฎีทำให้เห็นถึงการรวบรวมรายละเอียดที่สำคัญต่างๆ โดยมองในรูปทั่วๆ ไปเชิงนามธรรม ซึ่งทำให้รายละเอียดเหล่านี้กลายเป็นกรณีตัวอย่างเฉพาะและส่วนขยายของทฤษฎี

ทฤษฎีการคำนวณในหนังสือเล่มนี้ จะเน้นเกี่ยวกับความหมายของการคำนวณและภาวะการคำนวณได้ โดยการสร้างตัวแบบ (Model) ของเครื่องคอมพิวเตอร์ในลักษณะที่พิจารณาเฉพาะส่วนที่สำคัญของทั้งฮาร์ดแวร์และซอฟต์แวร์ ความเข้าใจในตัวแบบเหล่านี้จะช่วยให้ผู้เรียนสามารถเข้าใจว่า การคำนวณสามารถเข้าไปอยู่ในโปรแกรมที่ดำเนินงานบนเครื่องคอมพิวเตอร์จริงๆได้อย่างไร

มีหัวข้อหลายหัวข้อในทฤษฎีการคำนวณที่มีประโยชน์เชิงปฏิบัติที่สามารถเห็นได้ชัด เช่น ไวยากรณ์ไม่พึ่งบริบท (Context-Free Grammar) ซึ่งเป็นวิธีที่ใช้นิยามวากยสัมพันธ์ของภาษาโปรแกรมต่างๆ เพราะค่อนข้างง่ายต่อการเข้าใจ ไม่มีความซับซ้อน และ

ในขณะเดียวกันก็มีโปรแกรมที่จะแปลไวยากรณ์ไม่พึงบริบทเหล่านี้ให้เป็นออโตมาตากด ลง (Pushdown Automata) ซึ่งจะทำการวิเคราะห์การกระจายภาษาในตัวแปลภาษา (Compiler) หรือในตัวตีความภาษา (Interpreter) ส่วนหัวข้อเกี่ยวกับออโตมาตาจำกัด (Finite Automata) จะเป็นตัวแบบของการคำนวณอีกตัวแบบหนึ่ง ที่ถูกนำมาใช้ในตัวแปลภาษาและตัวตีความภาษา ตลอดจนใช้ในขั้นตอนวิธีจับคู่ (Pattern Matching Algorithm) อีกด้วย

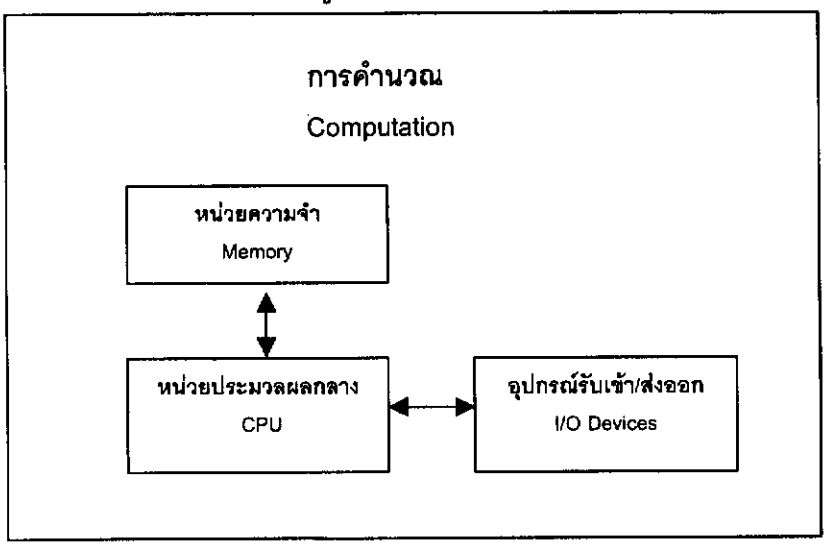
อีกเหตุผลหนึ่งที่แสดงถึงความสำคัญของทฤษฎีการคำนวณ คือ ตัวแบบเชิงทฤษฎี ที่ถูกรังสรรค์ขึ้นก่อนที่วัตถุเหล่านั้นจะถูกสร้างขึ้น ในความเป็นจริงที่เกิดขึ้นนี้ ตัวแบบเชิงทฤษฎีดังกล่าว จะเป็นเสมือนตัวดลใจและชักนำให้เกิดวัตถุเหล่านั้น เช่นเครื่องทัวริง (Turing Machine) ถูกทำให้เป็นรูปเป็นร่างขึ้นในลักษณะที่เป็นตัวแบบเชิงคณิตศาสตร์ของการคำนวณที่สามารถใช้ทำงานได้จริง ก่อนที่จะมีการผลิตเครื่องคอมพิวเตอร์ และเครื่องทัวริงนี้ก็เป็นแม่แบบสำหรับสถาปัตยกรรมของเครื่องคอมพิวเตอร์ชนิดที่มีหน่วยประมวลผลเดี่ยวที่ใช้กันอย่างแพร่หลายในปัจจุบัน

ดังนั้นการเรียนรู้ทฤษฎีจึงช่วยให้ผู้เรียนเข้าใจว่าทำไมฮาร์ดแวร์และซอฟต์แวร์จึงมีวิวัฒนาการที่ดียิ่งขึ้นในอนาคต และหวังว่าจากเหตุผลที่กล่าวมาทั้งหมดคงจะสามารถโน้มน้าวให้ผู้เรียนหรือผู้อ่านได้มองเห็นถึงประโยชน์ของการศึกษาทฤษฎีการคำนวณได้พอสมควร เมื่อผู้อ่านสามารถเข้าใจในทฤษฎีการคำนวณได้ในระดับหนึ่ง ผู้อ่านจะรู้สึกสนุกกับวิชานี้ และอาจพบว่าการสร้างเครื่องเล็กๆ ที่สามารถแก้ปัญหาต่างๆ หรือสร้างไวยากรณ์ที่ใช้บอกลักษณะของภาษาต่างๆ ก็สามารถให้ความเพลิดเพลินได้พอ ๆ กับการแก้ไขปริศนาได้เช่นกัน

II. ตัวแบบของการคำนวณ (Model of Computation)

ดังที่ได้กล่าวในหัวข้อที่แล้ว ทฤษฎีการคำนวณเน้นการศึกษาเกี่ยวกับตัวแบบของการคำนวณ และตัวแบบที่จะได้ศึกษาในหนังสือเล่มนี้ที่สำคัญๆ มีด้วยกัน 3 ตัวแบบ คือ ออโตมาตาจำกัด (Finite Automata) ออโตมาตากดลง (Pushdown Automata) และเครื่องทัวริง (Turing Machine) ตัวแบบทั้งสามนี้ถือเป็นออโตมาตาทั้งหมด ซึ่งต่างก็เป็นตัวแบบเชิงนามธรรมของเครื่องคอมพิวเตอร์

เป็นที่ทราบกันดีว่า เครื่องคอมพิวเตอร์ในปัจจุบันประกอบด้วยองค์ประกอบใหญ่ๆ 3 ส่วนคือ หน่วยประมวลผลกลาง (Central Processing Unit) หรือซีพียู (CPU) หน่วยความจำ (Memory) และอุปกรณ์รับเข้า/ส่งออก (Input/Output Unit Device หรือ อุปกรณ์ไอ/โอ (I/O Device)) ดังแสดงในรูป



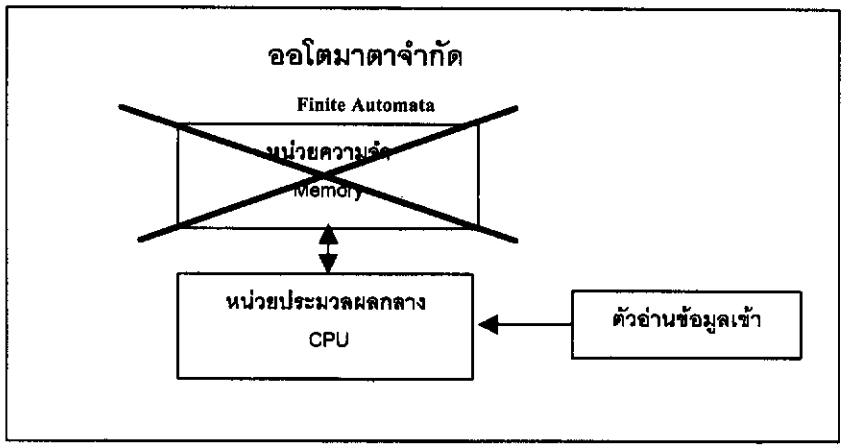
ซีพียู ทำหน้าที่ในการคำนวณทางคณิตศาสตร์และเลือกขั้นตอนการทำงานตามหลักตรรกะโดยขึ้นอยู่กับข้อมูล ขนาดของข้อมูลที่ซีพียูแต่ละอันจะสามารถรองรับได้ในขณะใดขณะหนึ่งจะถือว่าเป็นค่าตรึงถาวร ซึ่งขึ้นอยู่กับการออกแบบของเครื่องนั้นๆ ดังนั้นเพื่อให้เครื่องสามารถทำงานได้ในกรณีที่ขนาดของข้อมูลใหญ่กว่าค่าที่ตรึงไว้ จึงต้องมีการส่งผ่านข้อมูลกลับไปกลับมาระหว่างซีพียูกับ หน่วยความจำ และอุปกรณ์ไอ/โอ

หน่วยความจำ เป็นส่วนที่เก็บข้อมูลที่นอกเหนือจากที่ซีพียูจะสามารถเก็บได้รวมทั้งข้อมูลที่ซีพียูยังไม่ต้องการใช้ในขณะนั้น เพื่อแยกให้เห็นว่าส่วนนี้แตกต่างจากหน่วยความจำที่เป็นส่วนหนึ่งในซีพียู จึงมีการเรียกหน่วยความจำนี้อีกอย่างว่า หน่วยความจำช่วย (Auxiliary Memory) ในทางทฤษฎี ส่วนนี้เป็นส่วนที่ทำให้สามารถขยายความจุของพื้นที่เก็บได้อย่างไม่จำกัด

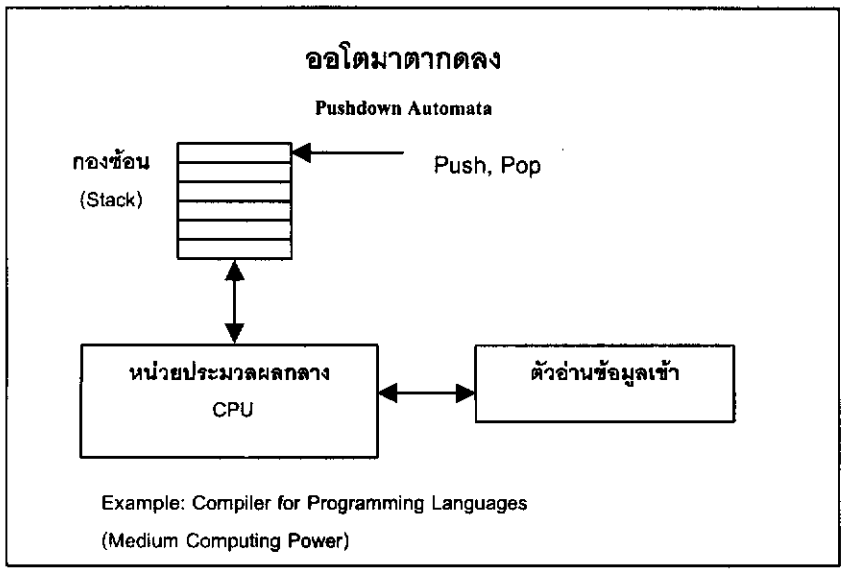
อุปกรณ์ไอ/โอ ทำหน้าที่เป็นตัวกลางระหว่างเครื่องคอมพิวเตอร์กับส่วนประกอบอื่นๆ รวมทั้งอุปกรณ์ต่อพ่วง (Peripheral Devices)

ถ้าพิจารณาตัวแบบทั้งสามที่จะได้เรียนในหนังสือเล่มนี้เปรียบเทียบกับเครื่องคอมพิวเตอร์ในปัจจุบันในเรื่องของส่วนประกอบที่ได้กล่าวมาข้างต้นนั้น จะสามารถสรุปได้ดังนี้

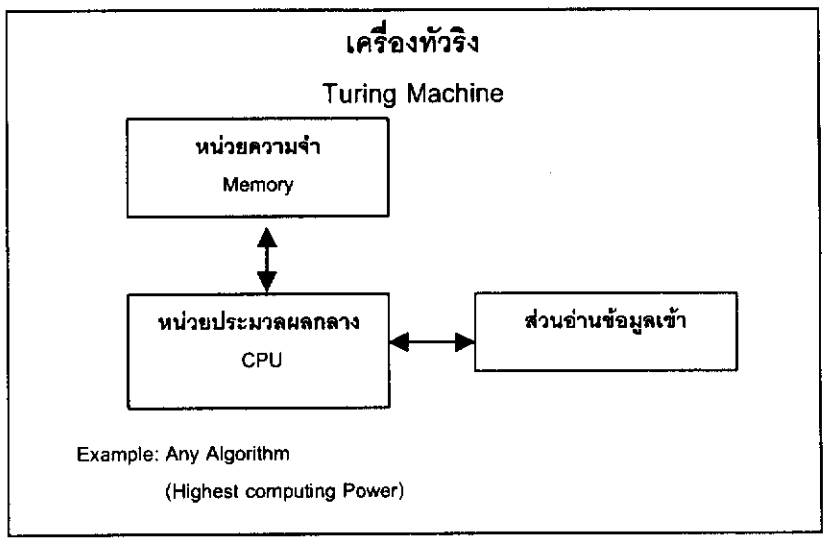
1. ออโตมาตาจำกัด จะเป็นเครื่องที่มีเฉพาะซีพียู และส่วนที่ทำหน้าที่อ่านข้อมูลเข้าเท่านั้น



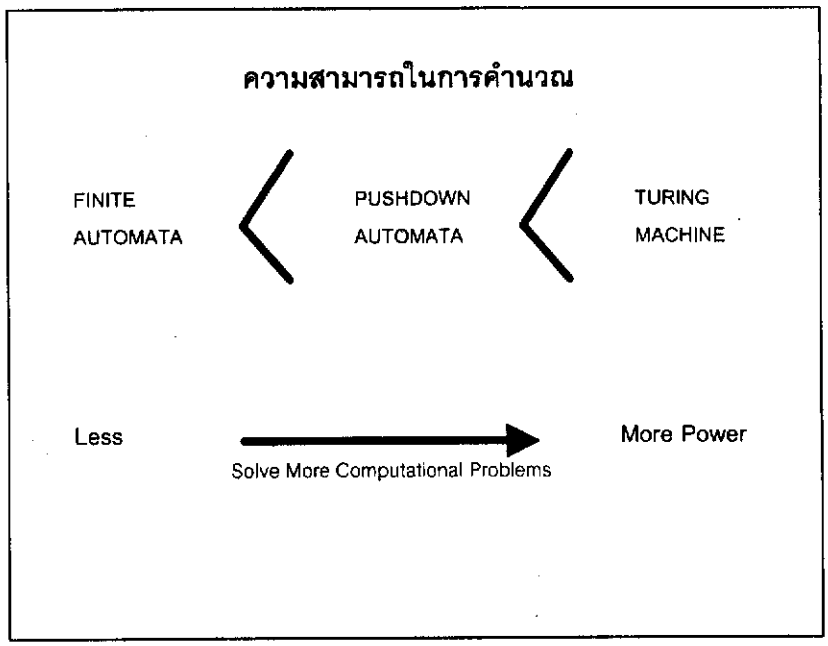
2. ออโตมาตากดลง จะเป็นเครื่องที่มีซีพียู หน่วยความจำช่วยแบบกองซ้อน (Stack) และส่วนที่ทำหน้าที่อ่านข้อมูลเข้า



3. เครื่องทัวริง จะเป็นเครื่องที่มีซีพียู หน่วยความจำ และส่วนที่ทำหน้าที่อ่านข้อมูลเข้า



จะเห็นว่าเครื่องทัวริงถือเป็นตัวแบบที่ใกล้เคียงกับเครื่องคอมพิวเตอร์ในปัจจุบันมากที่สุด โดยเปรียบเทียบความสามารถระหว่างตัวแบบทั้งสามได้ดังนี้



จากที่กล่าวมาทั้งหมดนี้พอจะสรุปถึงวัตถุประสงค์ในการเรียนวิชานี้ได้ดังนี้

1. เพื่อให้เข้าใจทฤษฎีบทพื้นฐานของการจดจำ (การวิเคราะห์) ซึ่งเป็นพื้นฐานของการออกแบบตัวแปรภาษา (Compiler)

2. เพื่อให้เข้าใจถึงตัวแบบ (Model) ของการคำนวณทั่ว ๆ ไปว่ามีจุดมุ่งหมายอย่างไร รวมถึงแสดงความสามารถของคอมพิวเตอร์ในรูปแบบของการแก้ไขปัญหาต่าง ๆ

3. ทำให้สามารถเข้าใจถึงตัวแบบเชิงนามธรรม (Abstract Model) ของกระบวนการการคำนวณเช่น ออโตมาตาจำกัด (Finite Automata) ออโตมาตากดลง (Pushdown Automata) และเครื่องทัวริง (Turing Machine)

4. ทำให้ผู้เรียนมีความสามารถในการเขียนและแก้ไขปัญหาต่าง ๆ ได้อย่างเข้าใจ ชัดเจน ถูกต้อง ด้วยการใช้สัญลักษณ์ทางคณิตศาสตร์ได้อย่างเหมาะสม

สำหรับเนื้อหาในหนังสือเล่มนี้มีการแบ่งออกเป็น 4 ส่วนดังนี้

ส่วนที่ 1 (Part 1): จะเป็นส่วนที่เกี่ยวข้องกับการกำหนดสัญลักษณ์และเทคนิคทางคณิตศาสตร์ (Mathematical Notation and Techniques) โดยจะกล่าวถึงเทคนิคที่ใช้ในการพิสูจน์ (Proof Techniques) การนิยามชุดตัวอักษร (Alphabet) สายอักขระ (String) และ ภาษา (Langugae) รวมไปถึงวิธีการนิยามแบบรีเคอร์ซีพ (Recursive Definition) เนื้อหาส่วนนี้ จะอยู่ในบทที่ 1

ส่วนที่ 2 (Part 2): จะเป็นส่วนที่เกี่ยวข้องกับภาษาปกติและออโตมาตาจำกัด (Regular Language and Finite Automata) โดยจะได้พิจารณาภาษาที่สนใจกลุ่มหนึ่งซึ่งเป็นกลุ่มภาษาที่อธิบายง่ายที่สุดนั่นคือ ภาษาปกติ จากนั้นจะได้พิจารณาเครื่องหรือตัวแบบที่ใช้ในการนิยามเพื่อยอมรับภาษาปกติดังกล่าวที่เรียกว่าออโตมาตาจำกัด ซึ่งถือว่าเป็นเครื่องคำนวณอย่างง่ายและไม่มีหน่วยความจำ นอกจากนี้ยังได้กล่าวถึงวิธีการพิจารณาภาษาอีกกลุ่มคือภาษาไม่ปกติ (Non-regular Language) เนื้อหาส่วนนี้จะอยู่ในบทที่ 2, 3, 4, และบทที่ 5

ส่วนที่ 3 (Part 3): จะเป็นส่วนที่เกี่ยวข้องกับภาษาไม่พึ่งบริบทและออโตมาตากดลง (Context-Free Languages and Pushdown Automata) โดยจะพิจารณาถึงกฎไวยากรณ์ที่ใช้สร้างภาษาด้วยวิธีรีเคอร์ซีพอย่างง่ายที่เรียกว่า ไวยากรณ์ไม่พึ่งบริบท

(Context-Free Grammars) รวมไปถึงเครื่องหรือตัวแบบที่สอดคล้องกับไวยากรณ์ไม่พึ่งบริบทดังกล่าว โดยเป็นเครื่องที่มีสถานะอย่างจำกัดแต่ได้เพิ่มหน่วยความจำเข้ามาที่เรียกว่ากองซ้อน (Stack) ซึ่งเรียกเครื่องหรือตัวแบบนี้ว่า ออโตมาตากดลง (Pushdown Automata) เนื้อหาส่วนนี้จะอยู่ในบทที่ 6 และ 7

ส่วนที่ 4 (Part 4): จะเป็นส่วนที่เกี่ยวข้องกับเครื่องหรือตัวแบบ ที่ได้รับการยอมรับโดยทั่วไป ว่าเป็นเครื่องต้นแบบของเครื่องคำนวณในปัจจุบัน (เครื่องคอมพิวเตอร์) ที่เรียกว่าเครื่องทัวริง (Turing Machines) เนื้อหาในส่วนนี้จะอยู่ในบทที่ 8