

แบบทดสอบต่อการเตรียมตัวสอบกลางภาค

ศิระกร ลำไย (5910500023) และกรวิชญ์ ชัยกังวาล (5910501909)

คำตอบที่มีการแยกช่อง คือแยกกันคิดวิเคราะห์ ก่อนจะร่วมกันอภิปรายผลลัพธ์

ข้อที่ 1

อยู่ในไฟล์ q1/q1.c

ข้อที่ 2

อยู่ในไฟล์ q2/q2.asm

ข้อที่ 3

อยู่ในไฟล์ q3/q3.asm

ข้อที่ 4

คำสั่งของ MOVN คือ if $rt \neq 0$ then $rd = rs$ และคำสั่งของ MOVZ คือ if $rt == 0$ then $rd = rs$ จะเห็นได้ว่าทั้งสองคำสั่งจำเป็นต้องมีการเปรียบเทียบเงื่อนไข ทำให้ datapath ของทั้งสองคำสั่งนั้นยุ่งยากขึ้นไปอีก นอกจากนั้นคำสั่งนี้จะต้องทำการ comparison และการ set ค่า ทำให้ต้องจัดการ clock ให้ข้อมูลไหลไปทัน

ข้อที่ 5

EBNF เป็น LL(1) grammar กล่าวคือ

- สามารถเลือกตัดสินใจที่จะ apply กฎของการ move ใดๆ ของ pushdown automata ด้วยการ “ส่อง” (peek) อักษรหนึ่งตัว
- เป็น grammar ที่สนใจ leftmost derivation ทำให้ได้ non ambiguous parsing

อาจสรุปได้ว่า “น้อมนำคำสอนของ LL(1) Language มาปฏิบัติเพื่อแก้ไขปัญหา parsing คืองานหลักของ EBNF”

ข้อที่ 6

เทียบได้กับ Regular Expression $\backslash d^+$ ซึ่งจะ capture เลข positive integer ออกมาใส่ในค่า val หลังจบการทำงาน

ข้อที่ 7

แกรมมาร์นั้นสามารถเขียนได้อยู่ในรูป $S \rightarrow \varepsilon \mid (S)S$
อยู่ในไฟล์ q7/q7.c

ข้อที่ 8

คำตอบของศิริกร	คำตอบของกรวิษฐ์
<p>ไม่สามารถทำนายอย่างแม่นยำได้ เนื่องจากในตัวนิสิตไม่มี branch predictor ที่แม่นยำระดับ 90% เหมือนกับ branch predictor บนซีพียูยุคปัจจุบัน</p> <p>แต่หากต้องทำนายจริงๆ ขอทำนายจาก course syllabus ว่าเป็นการใช้งานระบบ *NIX</p>	<p>เนื่องจากคราวที่แล้วทำตัว Parsing Tree และ จาก Course Syllabus แล้ว คาดว่าคราวหน้าจะมีการเขียน Compiler โดยใช้ Assembly หรือไม่ก็ อาจจะให้เรียนในหัวข้อ “ปฏิบัติการเกี่ยวกับการใช้งานระบบปฏิบัติการ UNIX” แต่เนื่องจากนิสิตส่วนใหญ่ นั้นมีพื้นฐานเกี่ยวกับ UNIX ในระดับหนึ่ง (มีหลายคนที่ใช้ Linux เป็น OS หลัก) เลยคาดว่าอาจจะมีการเรียนไปถึง “ปฏิบัติการเกี่ยวกับการใช้งาน thread ในสไตล์แบบ fork-join” ก็เป็นไปได้</p>