Final Exam

COMP – 520, Spring 2024

Please fill in today's date (YYYY-MM-DD)

_____-_____-_____

Write your name and PID here:

_____

Write your ALIAS here:

_____

Your alias and PID should be written on ALL exam pages.
Your alias represents you and will be equivalent to your name
throughout the exam.

Instructions

1. Open notes. (Anything you wrote down)
2. Open course website. (Including lectures)
3. Limited IDE access. (IntelliJ, Eclipse, Notepad, etc.)
   a. IDE access is given so that you can reliably see the syntax highlighting of your code, and that's it!
   b. You may open your IDE but not write any new code.
   c. You may NOT compile any code. (javac)
   d. You may NOT run anything in your IDE.
4. No websites other than the course website/gradescope.
   a. No Piazza, Canvas, Sakai, email, etc.
   b. You **MAY** use the web resources specified in PA4.
5. No phones at all, even if you forgot your laptop.
6. No collaboration. No communication with anyone other than instructors, TAs, and proctors.

7. Be concise in your answers.
8. You get the full exam duration (180 minutes).
9. Sign below with either your name or alias.

I pledge I have not given nor received unauthorized aid on this exam:

_____

PLEASE READ THIS PAGE ON GRADING

You only need 100 points to get a 100% on this exam.

Note: there are 5 questions each worth 25 points.

This means 125 points are possible.

Any excess of 100 points will be applied as extra credit with the weight of a midterm.

Cheat Sheet

Dataflow Analysis:

$$\text{for}(\ v \in V\ )\ \{$$
$$\text{out}(v) := \bigcup s\colon s \in \text{successor}(v) :: \text{in}(s)$$
$$\text{in}(v) := \text{use}(v) \cup \big(\text{out}(v) \setminus def(v)\big)$$
$$\}$$

Invariants:

$$I0 \equiv\ \text{use}(v) \subseteq \text{in}(v)$$
$$I1 \equiv\ \text{out}(v) \setminus \text{def}(v) \subseteq \text{in}(v)$$
$$I2 \equiv\ \forall\ s : s \in \text{successor}(v) :: \text{in}(s) \subseteq \text{out}(v)$$

Notation:

Followers(A) $\equiv$ FL(A)

Starters(A) $\equiv$ ST(A)

Nullable(A) $\equiv$ N(A)

Capitals are non-terminals, lowercase are terminals.

Greek letters are sequences.

Nullable Induction:

Observed	Rule
1. $\alpha = \varepsilon$	Nullable(α) = true
2. $\alpha = t$	Nullable(α) = false
3. $\alpha = A$	Nullable(α) = Nullable(A)
4. $\alpha = \alpha_1 \alpha_2 \ldots \alpha_n$	Nullable(α) = Nullable(α_1) $\wedge \ldots \wedge$ Nullable(α_n)
5. $\alpha = \alpha_1 \vert \alpha_2 \vert \ldots \vert \alpha_n$	Nullable(α) = Nullable(α_1) $\vee \ldots \vee$ Nullable(α_n)
6. $\alpha = \beta^*$	Nullable(α) = true

Starters Induction:

Observed	Rule
1. $\alpha = \varepsilon$	Starters(α) = { ε }
2. $\alpha = t$	Starters(α) = { t }
3. $\alpha = A$	Starters(α) = Starters(A)
4. $\alpha = \alpha_1\alpha_2 \dots \alpha_n$	Starters(α) = Starters(α_1)$\oplus$Starters($\alpha_2 \dots \alpha_n$)
5. $\alpha = \alpha_1\|\alpha_2\| \dots \|\alpha_n$	Starters(α) = Starters(α_1) $\cup$ … $\cup$ Starters(α_n)
6. $\alpha = \beta^*$	Starters(α) = Starters(β) $\cup$ { ε }

$$A \oplus B = \begin{cases} A & \text{if } \varepsilon \notin A \\ (A\backslash\{\varepsilon\}) \cup B & \text{otherwise} \end{cases}$$

Followers First Step:

$$FL_0(A) = \left(\bigcup_{C::=\alpha A\beta} ST(\beta) \right) \backslash\{\varepsilon\}$$

Followers Inductive Step:

$$FL_{i+1}(A) = FL_i(A) \cup \bigcup_{C::=\alpha A\beta \text{ and Nullable}(\beta)} FL_i(C)$$

Followers Final Step:

$$FL(A) = FL_n(A) \cup \begin{cases} \{\varepsilon\} & \text{if } S \Rightarrow^* \alpha A \\ \{\} & \text{otherwise} \end{cases}$$

For each choice in A ::= $\beta(\alpha_1 | \ldots | a_m)\gamma$ define:

    Predict(α_i) = Starters($\alpha_i\gamma$) $\oplus$ Followers(A)

For repetitions in A ::= $\beta(\alpha)^*\gamma$

    Predict(α) = Starters(α)

    Predict(γ) = Starters(γ) $\oplus$ Followers(A)

For sequences A ::= $\alpha\beta\gamma$

    Predict(A) = ST(α) $\oplus$ ST(β) $\oplus$ ST(γ)

x64:

    pop rm: Stores data at [rsp] into rm, and rsp += 8

    push rm: Does rsp -= 8, then stores rm at [rsp]

    mov rm,r: Takes r, stores it in rm

    mov rm,imm32: Stores imm32 in rm

RM examples:

Form	Example
r	# rdx
[rdisp + disp]	# [rbp-10]
[ridx*mult + disp]	# [rcx*8+11223344]
[rdisp + ridx*mult + disp]	# [rbx+rsi*4-8]
[disp]	# [11223344]

Useful Registers:

    rax, rcx, rdx, rbx, rsp, rbp, rsi, rdi

    r8, r9, r10, r11, r12, r13, r14, r15