CS415 Compilers: First Recitation

January 28, 2022
piazza.com

I’ll be checking this
piazza.com

I’ll be checking this

Office Hours

3pm to 4pm EST on Thursdays
It’s at this same Zoom location
You can come just to do work, even if you don’t have any questions at all.
piazza.com
   I’ll be checking this
Office Hours
   3pm to 4pm EST on Thursdays
   It’s at this same Zoom location
   You can come just to do work, even if you don’t have any questions at all.
I do not use Canvas Inbox
Running ILOC

Connect to ilab.cs.rutgers.edu
Running ILOC

Connect to ilab.cs.rutgers.edu

How many have not done this? Please say so in the chat.
Running ILOC

Connect to ilab.cs.rutgers.edu

How many have not done this? Please say so in the chat.

Get and test the executable for the simulator

```bash
mkdir cs415
cd cs415
cp -r /common/home/uli/cs415/ILOC_Simulator .
cd ILOC_Simulator
./sim < test.i
```

Who is having difficulties?

CS415 Compilers: First Recitation January 28, 2022 3 / 14
Running ILOC

Connect to ilab.cs.rutgers.edu
   How many have not done this? Please say so in the chat.

Get and test the executable for the simulator
   mkdir cs415
   cd cs415
   cp -r /common/home/uli/cs415/ILOC_Simulator .
   cd ILOC_Simulator
   ./sim < test.i
   less ReadMe  # Take the time later on to read this.
Running ILOC

Connect to ilab.cs.rutgers.edu
   How many have not done this? Please say so in the chat.
Get and test the executable for the simulator
   mkdir cs415
   cd cs415
   cp -r /common/home/uli/cs415/ILOC_Simulator .
   cd ILOC_Simulator
   ./sim < test.i
   less ReadMe  # Take the time later on to read this.
Who is having difficulties?
Precedence graph

We will assign priorities based on longest latency-weighted path.
We will assign priorities based on longest latency-weighted path.
(See lecture slides for other possible priorities)
Precedence graph

We will assign priorities based on longest latency-weighted path.
(See lecture slides for other possible priorities)

Operation: Cycles

- add: 1
- cmp_LE: 2
- div: 2
- fact: 4
- i2i: 1
- load: 3
- loadI: 1
- loadAI: 3
- output: 1
- outputAI: 1
- store: 3
- storeAI: 3
- shift: 1
a) loadI 0 => r1
b) loadI 0 => r2
c) i2i r1 => r3
d) addI r1, 1 => r1
e) fact r3 => r4
f) loadI 100000 => r1
g) div r1, r4 => r3
h) add r3, r2 => r2
i) loadI 0 => r5
j) cmp_LE r3, r5 => r5
k) storeAI r2 => r0, 4
a) loadI 0 => r1
b) loadI 0 => r2
c) i2i r1 => r3
d) addI r1, 1 => r4
e) i2i r4 => r1
f) fact r3 => r5
g) loadI 100000 => r6
h) div r6, r5 => r7)i) add r7, r2 => r2
j) loadI 0 => r8
k) cmp_LE r7, r8 => r9
l) storeAI r2 => r0, 4
Draw the dependence graph in breakout rooms

a) loadI 0 => r1
b) loadI 0 => r2
c) i2i r1 => r3
d) addI r1, 1 => r4
e) i2i r4 => r1
f) fact r3 => r5
g) loadI 100000 => r6
h) div r6, r5 => r7
i) add r7, r2 => r2
j) loadI 0 => r8
k) cmp_LE r7, r8 => r9
l) storeAI r2 => r0, 4

add: 1
cmp_LE: 2
div: 2
fact: 4
i2i: 1
load: 3
loadI: 1
loadAI: 3
output: 1
outputAI: 1
store: 3
storeAI: 3
shift: 1
a) `loadI 0 => r1`
b) `loadI 0 => r2`
c) `i2i r1 => r3`
d) `addI r1, 1 => r4`
e) `i2i r4 => r1`
f) `fact r3 => r5`
g) `loadI 100000 => r6`
h) `div r6, r5 => r7`
i) `add r7, r2 => r2`
j) `loadI 0 => r8`
k) `cmp_LE r7, r8 => r9`
l) `storeAI r2 => r0, 4`
a) \texttt{loadI 0} => r1
b) \texttt{loadI 0} => r2
c) \texttt{i2i r1} => r3
d) \texttt{addI r1, 1} => r4
e) \texttt{i2i r4} => r1
f) \texttt{fact r3} => r5
g) \texttt{loadI 100000} => r6
h) \texttt{div r6, r5} => r7
i) \texttt{add r7, r2} => r2
j) \texttt{loadI 0} => r8
k) \texttt{cmp_LE r7, r8} => r9
l) \texttt{storeAI r2} => r0, 4
Dependence
Dependence: ALSU Exercise 10.2.1

For each of the stated pairs, determine whether it has an true dependence, antidependence, output dependence, or none at all.

\[
\begin{align*}
    a &= b \\
    c &= d \\
    b &= c \\
    d &= a \\
    c &= d \\
    a &= b
\end{align*}
\]

1, 4

3, 5

1, 6

3, 6

4, 6
For each of the stated pairs, determine whether it has an true dependence, antidependence, output dependence, or none at all.

\[
\begin{align*}
a &= b & (1, 4) & \text{True dependence} \\
c &= d & \quad & \text{3, 5} \\
b &= c \\
d &= a \\
c &= d \\
a &= b
\end{align*}
\]
For each of the stated pairs, determine whether it has an true dependence, antidependence, output dependence, or none at all.

\[
\begin{align*}
\text{a} & = \text{b} \\
\text{c} & = \text{d} \\
\text{b} & = \text{c} \\
\text{d} & = \text{a} \\
\text{c} & = \text{d} \\
\text{a} & = \text{b}
\end{align*}
\]

1, 4 True dependence
3, 5 Antidependence
1, 6
For each of the stated pairs, determine whether it has an true dependence, antidependence, output dependence, or none at all.

\[
\begin{align*}
a &= b & \quad & 1, 4 \text{ True dependence} \\
c &= d & \quad & 3, 5 \text{ Antidependence} \\
b &= c & \quad & 1, 6 \text{ Output dependence} \\
d &= a & \quad & 3, 6 \\
c &= d & \\
a &= b
\end{align*}
\]
For each of the stated pairs, determine whether it has an true dependence, antidependence, output dependence, or none at all.

\[
\begin{align*}
a &= b & & 1, 4 \text{ True dependence} \\
c &= d & & 3, 5 \text{ Antidependence} \\
b &= c & & 1, 6 \text{ Output dependence} \\
d &= a & & 3, 6 \text{ True dependence} \\
c &= d & & 4, 6 \\
a &= b & &
\end{align*}
\]
For each of the stated pairs, determine whether it has an true dependence, antidependence, output dependence, or none at all.

a = b  
1, 4 True dependence

c = d  
3, 5 Antidependence

b = c  
1, 6 Output dependence

d = a  
3, 6 True dependence

c = d  
4, 6 Antidependence

a = b
load r1 => r11 // r11 = r1
store r12 => r1 // r12 = t1
load r2 => r13 // r2 = r13
store r14 => r2 // r14 = r2

Are there any read-after-write dependencies (true dependence)?
No
Are there any write-after-read dependencies (antidependence)?
Yes
Instruction 2 antidependent on instruction 1
Instruction 4 antidependent on instruction 3
Are there any write-after-write dependencies (output dependence)?
No.
load r1 => r11 // r11 = r1
store r12 => r1 // r12 = t1
load r2 => r13 // r2 = r13
store r14 => r2 // r14 = r2

Are there any read-after-write dependencies (true dependence)?

No

Are there any write-after-read dependencies (antidependence)?

Yes
Instruction 2 antidependent on instruction 1
Instruction 4 antidependent on instruction 3

Are there any write-after-write dependencies (output dependence)?

No
Dependence: ALSU Example 10.2

load r1 => r11 // r11 = r1
store r12 => r1 // r12 = t1
load r2 => r13 // r2 = r13
store r14 => r2 // r14 = r2

Are there any read-after-write dependencies (true dependence)? No
Are there any write-after-read dependencies (antidependence)?
Dependence: ALSU Example 10.2

\[
\begin{align*}
\text{load } r_1 & \Rightarrow r_{11} \quad // \quad r_{11} = r_1 \\
\text{store } r_{12} & \Rightarrow r_1 \quad // \quad r_{12} = t_1 \\
\text{load } r_2 & \Rightarrow r_{13} \quad // \quad r_2 = r_{13} \\
\text{store } r_{14} & \Rightarrow r_2 \quad // \quad r_{14} = r_2
\end{align*}
\]

Are there any read-after-write dependencies (true dependence)? **No**

Are there any write-after-read dependencies (antidependence)? **Yes**
Dependence: ALSU Example 10.2

load r1 => r11 // r11 = r1
store r12 => r1 // r12 = t1
load r2 => r13 // r2 = r13
store r14 => r2 // r14 = r2

Are there any read-after-write dependencies (true dependence)? **No**
Are there any write-after-read dependencies (antidependence)? **Yes**

Instruction 2 antidependent on instruction 1
load r1 => r11 // r11 = r1
store r12 => r1 // r12 = t1
load r2 => r13 // r2 = r13
store r14 => r2 // r14 = r2

Are there any read-after-write dependencies (true dependence)? No
Are there any write-after-read dependencies (antidependence)? Yes

Instruction 2 antidependent on instruction 1
Instruction 4 antidependent on instruction 3
load r1 => r11 // r11 = r1
store r12 => r1 // r12 = t1
load r2 => r13 // r2 = r13
store r14 => r2 // r14 = r2

Are there any read-after-write dependencies (true dependence)? No
Are there any write-after-read dependencies (antidependence)? Yes
  Instruction 2 antidependent on instruction 1
  Instruction 4 antidependent on instruction 3

Are there any write-after-write dependencies (output dependence)?
dependence: ALSU example 10.2

load r1 => r11 // r11 = r1
store r12 => r1 // r12 = t1
load r2 => r13 // r2 = r13
store r14 => r2 // r14 = r2

are there any read-after-write dependencies (true dependence)? no
are there any write-after-read dependencies (antidependence)? yes

   Instruction 2 antidependent on instruction 1
   Instruction 4 antidependent on instruction 3

are there any write-after-write dependencies (output dependence)? no.
(a+b)+c+(d+e)

load r1 => a
load r2 => b
add r1, r1 => r2
load r2, c
add r1, r1 => r2
load r2 => d
load r3 => e
add r2, r2 => r3
add r1, r1 => r2

Are there any read-after-write dependencies (true dependence)?

Are there any write-after-read dependencies (antidependence)?

Are there any write-after-write dependencies (output dependence)?
(a+b)+c+(d+e)

load r1 => a
load r2 => b
add r1, r1 => r2
load r2, c
add r1, r1 => r2
load r2 => d
load r3 => e
add r2, r2 => r3
add r1, r1 => r2

Are there any read-after-write dependencies (true dependence)? **No**
(a+b)+c+(d+e)

load r1 => a
load r2 => b
add r1, r1 => r2
load r2, c
add r1, r1 => r2
load r2 => d
load r3 => e
add r2, r2 => r3
add r1, r1 => r2

Are there any read-after-write dependencies (true dependence)? **No**

Are there any write-after-read dependencies (antidependence)?
Dependence: From ALSU Example 10.2

\[(a+b)+c+(d+e)\]

load r1 => a
load r2 => b
add r1, r1 => r2
load r2, c
add r1, r1 => r2
load r2 => d
load r3 => e
add r2, r2 => r3
add r1, r1 => r2

Are there any read-after-write dependencies (true dependence)?  **No**
Are there any write-after-read dependencies (antidependence)?  **Yes**
Dependence: From ALSU Example 10.2

(a+b)+c+(d+e)

load r1 => a
load r2 => b
add r1, r1 => r2
load r2, c
add r1, r1 => r2
load r2 => d
load r3 => e
add r2, r2 => r3
add r1, r1 => r2

Are there any read-after-write dependencies (true dependence)? No

Are there any write-after-read dependencies (antidependence)? Yes

Are there any write-after-write dependencies (output dependence)?
(a+b)+c+(d+e)

load r1 => a
load r2 => b
add r1, r1 => r2
load r2, c
add r1, r1 => r2
load r2 => d
load r3 => e
add r2, r2 => r3
add r1, r1 => r2

Are there any read-after-write dependencies (true dependence)? **No**
Are there any write-after-read dependencies (antidependence)? **Yes**
Are there any write-after-write dependencies (output dependence)? **No.**
piazza.com
I’ll be checking this
piazza.com

I’ll be checking this

Office Hours

3pm to 4pm EST on Thursdays
It’s at this same Zoom location
You can come just to do work, even if you don’t have any questions at all.
piazza.com
    I’ll be checking this
Office Hours
    3pm to 4pm EST on Thursdays
    It’s at this same Zoom location
    You can come just to do work, even if you don’t have any questions at all.
I do not use Canvas Inbox