CS415 Compilers: First Recitation

January 28, 2022

Administrivia

- piazza.com
  - I'll be checking this

Administrivia

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.

Administrivia

- 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

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

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.

Precedence graph

We will assign priorities based on longest latency-weighted path.

(See lecure slides for other possible priorities)

Precedence graph

We will assign priorities based on longest latency-weighted path.

(See lecure 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`

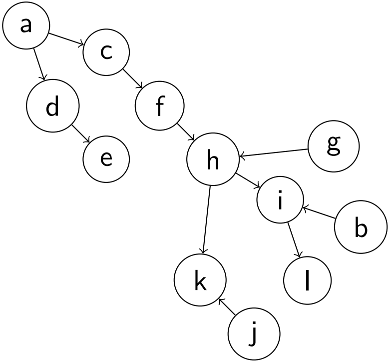
a) `loadI 0 => r1`
b) `loadI 0 => r2`
c) `i2i r1 => r3`
d) `addI r1, 1 => r4`
e) `i2i r4 => r1`
f) `fact r3 => r5`
g) `loadI 100000 => r6`
h) `div r6, r5 => r7`
i) `add r7, r2 => r2`
j) `loadI 0 => r8`
k) `cmp_LE r7, r8 => r9`
l) `storeAI r2 => r0, 4`

Draw the dependence graph in breakout rooms

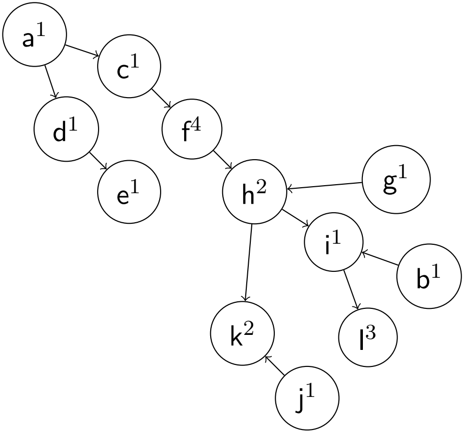
a) `loadI 0 => r1`
b) `loadI 0 => r2`
c) `i2i r1 => r3`
d) `addI r1, 1 => r4`
e) `i2i r4 => r1`
f) `fact r3 => r5`
g) `loadI 100000 => r6`
h) `div r6, r5 => r7`
i) `add r7, r2 => r2`
j) `loadI 0 => r8`
k) `cmp_LE r7, r8 => r9`
l) `storeAI r2 => r0, 4`

`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) 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



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.

```
a = b
c = d
b = c
d = a
c = d
a = b
```

1, 4

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.

```
a = b
c = d
b = c
d = a
c = d
a = b
```

1, 4 True dependence

3, 5

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.

```
a = b
c = d
b = c
d = a
c = d
a = b
```

1, 4 True dependence

3, 5 Antidependence

1, 6

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.

```
a = b
c = d
b = c
d = a
c = d
a = b
```

1, 4 True dependence

3, 5 Antidependence

1, 6 Output dependence

3, 6

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.

```
a = b
c = d
b = c
d = a
c = d
a = b
```

1, 4 True dependence

3, 5 Antidependence

1, 6 Output dependence

3, 6 True dependence

4, 6

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.

```
a = b
c = d
b = c
d = a
c = d
a = b
```

1, 4 True dependence

3, 5 Antidependence

1, 6 Output dependence

3, 6 True dependence

4, 6 Antidependence

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

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)?

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

```
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**

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

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**

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)?

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.**

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)?

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**

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)?

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)?

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)? **No.**

Administrivia

- piazza.com
  - I'll be checking this

Administrivia

- 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.

Administrivia

- 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