CSE443
Compilers

Dr. Carl Alphonce
alphonce@buffalo.edu
343 Davis Hall
Project 2
due date adjustment

Wednesday March 13 @ 5:00 PM

What hasn’t changed

Homework 2 still due today @ 5:00
Homework 3 released today
Project 3 released today
Example from last class

```plaintext
S -> while ( 
    { L1 = new ();
      L2 = new();
      C.false = S.next;
      C.true = L2;
    }
  )
C )

{ S1.next = L1; }
S1

{ S.code = label || L1 || C.code ||
    label || L2 || S1.code; }
```
Example from last class

\[ S \rightarrow \text{while (} \]
\[ \quad M \quad C \quad ) \]
\[ N \quad S1 \]
\[ \quad \{ \quad S\text{.code} = \text{label} \\| \quad L1 \\| \quad C\text{.code} \\| \]
\[ \quad \quad \quad \text{label} \\| \quad L2 \\| \quad S1\text{.code}; \quad \} \]
\[ M \rightarrow \epsilon \]
\[ \quad \{ \quad L1 = \text{new} (); \]
\[ \quad \quad L2 = \text{new}(); \]
\[ \quad \quad C\text{.false} = S\text{.next}; \]
\[ \quad \quad C\text{.true} = L2; \]
\[ \quad \} \]
\[ N \rightarrow \epsilon \]
\[ \quad \{ \quad S1\text{.next} = L1; \quad \} \]
Phases of a compiler

Intermediate Representation (IR): specification and generation

Figure 1.6, page 5 of text
HLL 1 -> IR -> target 1
HLL 2 -> IR -> target 2
HLL m -> IR -> target n
Machine independent optimizations

Machine dependent optimizations

IR → IR → Target 1 → Target 2 → Target n
Intermediate Representations
Directed Acyclic Graph (DAG)

- Similar to a syntax tree
- No repeated nodes: structure sharing
Ex. 6.1 [p 359]

\[ a + a \times (b - c) + (b - c) \times d \]
Ex. 6.1 [p 359]

\[ a + a \ast (b - c) + (b - c) \ast d \]
Ex. 6.1 [p 359]

\[ a + a \times (b - c) + (b - c) \times d \]
Ex. 6.1 [p 359]

$$a + a \ast (b - c) + (b - c) \ast d$$

Things can be more complicated if expressions have side effects.
Three address code instructions
(see 6.2.1, pages 364-5)

1. \( x = y \text{ op } z \)  
2. \( x = \text{ op } y \) \hspace{1cm} (treat i2r and r2i as unary ops)
3. \( x = y \)  
4. goto L  
5. if \( x \) goto L / ifFalse \( x \) goto L  
6. if \( x \) relop \( y \) goto L  
7. function calls:
   - param \( x \)  
   - call \( p, n \)  
   - \( y = \text{ call } p \)  
   - return \( y \)  
8. \( x = y[i] \) and \( x[i] = y \)  
9. \( x = &y, x = *y, *x = y \)