# CS241 Midterm Review Session

Graham Cooper

June 24th, 2015

## Summary of Topics

- Bits and Bytes
- MIPS Machine Code
- Assembly Language Programming
- MERL and Linking
- Regular Languages
- Context-Free Grammar

## Bits, Bytes and Words

Range of 2's compliment $[-2^{k-1}, 2^{k-1} - 1]$
Unsigned Integers range $[0, 2^k - 1]$

-25 to an 8-bit two's compliment
00011001
flip the bits
11100110
00000001
———
11100111

## Assembly Programming

Input \$1 = n ($n \leq n \leq 10$)
Output: \$3 = n!

```
factorial:
sw $2, -4($30)
sw $4, -8($30)
```

```
lis $2
.word 8
sub $30, $30, $2
add $2, $1, $0
lis $4
.word $1
add $3, $0, $4
loop: mult $3, $2
mflo $3
sub $2, $2, $4
bne $2, $0, loop
lis $2
.word 8
add $30, $30, $2
lw $2, -4($30)
lw $4, -8($30)
jr $31
```

# Assembly Programs

Hex notation for:

```
beq $20, $13, -4
```

0001 00ss ssst tttt iiii iiii iiii iiii
→
0001 0010 1000 1101 1111 1111 1111 1100

# MERL Linking

**HEADER** - Cookie - MERL file length - Code Len + header **MIPS CODE**
**HEADER** - Relocation (.word with label) - - 0x01 - ESR (External Symbol Reference) .import - - 0x11 - ESD (External Symbol Definition) .export - - 0x05
**File A**

```
.import abc
.word 0x18
.word abc
```

**File B**

```
.export abc
.export def
.word def
abc: def:
```

**Linked MERL File**

```
.word 0x10000002
.word filelen
.word codelen
.word 0x18
.word abc
.word def
abc: def:
.word 0x01
.word 0x10
.word 0x01
.word 0x14
.word 0x05
.word 0x18
.word 0x3
.word ascii a
.word ascii b
.word ascii c
.word 0x05
.word 0x18
.word 0x3
.word ascii d
.word ascii e
.word ascii f
```

# Regular Languages

## DFA

- $\Sigma$ - Alphabet, set of symbols we can use to form words
- Q - Finite set of states
- $q_0$ - Starting State
- A - finite set of Accepting States
- $\delta$: Q $\times$ $\Sigma$ $\rightarrow$ Q - transition function)

## NFA

- $\Sigma$ - Alphabet, set of symbols we can use to form words
- Q - Finite set of states
- $q_0$ - Starting State
- A - finite set of Accepting States
- $\delta$: Q $\times$ $\Sigma$ $\rightarrow$ $2^Q$ - transition function)

## Example:

Construct a DFA:
$\Sigma$ = {0,1,2,3,4}
L1 = {even integers with no useless leading zeros}
Construct an NFA:
$\Sigma$ = {a,b,c,d}
L2 = {words that contain "cab" as a substring and end in "dad"}

## $\epsilon$-NFA

$\Sigma$ = {a,b,c,d,0,1,2,3,4}
L = L1 and L2

### Regular Expressions Examples

$\Sigma$ = {a,b,c}
L = {words with an even number of a's}

(b|c|a(b|c)*a)*

$\Sigma$ = {a,b,c}
L = {all strings with 3 consectutive a's, entually followed by 4 consecutive b's}

(a|b|c)*aaa(a|b|c)*bbbb(a|b|c)*

## Context Free Grammar

- V - Nonterminals (set)
- $\Sigma$ - Terminals (set)
- R - Production Rules (set)
- S - Starting nonterminal

S$\rightarrow$ AyxxA | xA
A $\rightarrow$ Ax |Ay| $\epsilon$
S $\implies$ xA $\implies$