# Principles of Programming Languages

CS 314

Recitation 1

## Topics Today

Rewrite System

Regular Expressions

# Rewrite System-Example

Input: Sequence of characters starting with $ and ending with #, and any combination of 0s and 1s in between.

Rules: You may replace a character pattern X at any position within the character sequence on the left-hand-side by the pattern Y on the right-hand-side: X => Y

## Rewrite System-Example

Rewrite $01101# and $10100# use the following rules

- rule 1 : $1 => 1&
- rule 2 : $0 => 0$
- rule 3 : &1 => 1$
- rule 4 : &0 => 0&
- rule 5 : $# => A
- rule 6 : &# => B

RUTGERS

## Rewrite System-Example

Input: $01101#

rule2: 0$1101#

rule1: 01&101#

rule3: 011$01#

rule2: 0110$1#

rule1: 01101&#

rule6: 01101B

- rule 1 : $1 => 1&
- rule 2 : $0 => 0$
- rule 3 : &1 => 1$
- rule 4 : &0 => 0&
- rule 5 : $# => A
- rule 6 : &# => B

RUTGERS

## Rewrite System-Example

Input: $10100#

rule1: 1&0100#

rule4: 10&100#

rule3: 101$00#

rule2: 1010$0#

rule2: 10100$#

rule5: 10100A

- rule 1 : $1 => 1&
- rule 2 : $0 => 0$
- rule 3 : &1 => 1$
- rule 4 : &0 => 0&
- rule 5 : $# => A
- rule 6 : &# => B

## Rewrite System-Question

Is the output always unique? Can we have various output by applying rules in different orders?

RUTGERS

## Rewrite System-Example

Input: $10100#

- rule 1 : $1 => 1&
- rule 2 : $0 => 0$
- rule 3 : &1 => 1$
- rule 4 : &0 => 0&
- rule 5 : $# => A
- rule 6 : &# => B

- extra rule 7: $10 => $

$10100# => 1&0100# (rule1)

$10100# => $100# (rule7)

- extra rule 8: 1& => $

1&0100# => 10&100# (rule4)

1&0100# => $0100# (rule8)

## Rewrite System-Question

Is it possible that the system fails to give an output?

## Rewrite System-Example

Input: $10100#

- rule 1 : 01 => 10
- rule 2 : 10 => 01
- rule 3 : &1 => 1$
- rule 4 : &0 => 0&
- rule 5 : $# => A
- rule 6 : &# => B

Keep applying rule 1 and 2, the procedure cannot stop.

RUTGERS

## Regular Expressions

A syntax (notation) to specify regular languages.

It is one of the following:

1. A character
2. Empty String ($\in$)
3. one regular expression followed by another regular expression
4. regular expression 1|regular expression 2
5. regular expression*
6. regular expression$^+$

## Regular Expressions-Example

Construct a regular expression for binary numbers with even length

((1|0)(1|0))*

RUTGERS

## Regular Expressions-Example

Construct a regular expression for binary numbers with odd length

$(1|0)((1|0)(1|0))*$

## Regular Expressions-Example

All strings of a’s, b’s, and c’s that contain at least 2a’s.

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

## Regular Expressions-Example

All strings of a’s, b’s, and c’s that contain no consecutive a’s.

$(\in|a)((b|c)^{+}(\in|a))^{*}$

$(b|c)^{*}((a(b|c)^{+})|(b|c))^{*}(\in|a)$

## Regular Expressions-Example

Variables in C can contain lowercase characters, uppercase characters, numbers, and underscores, but can’t start with a number.

lower=a|b|c|...|z

upper=A|B|...|Z

number=1|2|...|9

## Regular Expressions-Example

Variables in C can contain lowercase characters, uppercase characters, numbers, and underscores, but can't start with a number.

(_|lower|upper)(_|lower|upper|number)*

## Regular Expressions-Example

Construct a regular expression for floating point numbers that don’t use scientific notation (e.g., 3.5, 0.15, -47.3).

$(- | \in)(0-9)^{+}.(0-9)^{+}$