Università
della
Svizzera
italiana

Facoltà
di scienze
informatiche

Theory of Computation
Supplementary math session

Spring 2017

Professor Natasha Sharygina
Informatics Department
University of Lugano
www.inf.unisi.ch

Math session

- Preliminaries
  - Set Notation, Strings, Languages, Graphs
  - First order logic
  - Propositional logic

Sets, Strings, Languages, Graphs

- *Set* – collection of finite or infinite elements

Examples: $A=\{1,5,7,10\}$, $\{x|R(x)\}$, $\{\emptyset\}$

$|A|$ - number of elements (size or cardinality) of set A

Operations:

$A\cap B$, $A\cup B$, $A\setminus B$, 2^A, $A \in D$, $D = \{\dots, -1, 0, 1, 2, \dots\}$ - *Int*
or $D = \{0,1,2,\dots\}$ – *Nat*, $A \notin D$, $A \subseteq B$, $A = B$, $A \subset B$

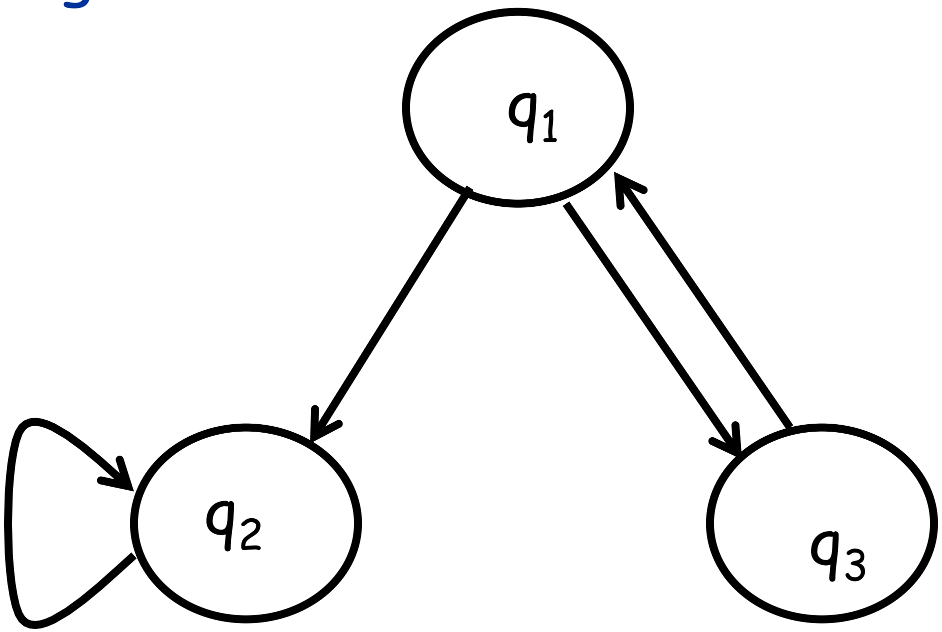
Cartesian product of A and B: $A \times B$ – set of ordered pairs
n-tuple is an object with exactly n elements

Sets, Strings, Languages, Graphs

- String – finite (infinite) sequence over some pre-defined alphabet (Σ), ε - empty string
- Language (L) – finite (infinite) set of strings over Σ
- Operations on strings:
  - Concatenation (abba.baa.cbb)
  - Repetition (σ^* - set of strings with a finite repetition of σ) L* - language obtained from L by repeatedly concatenating 0 or more words from L
  - Set operators
  - u.v – u is prefix, v is suffix
  - What is n-fix notation?

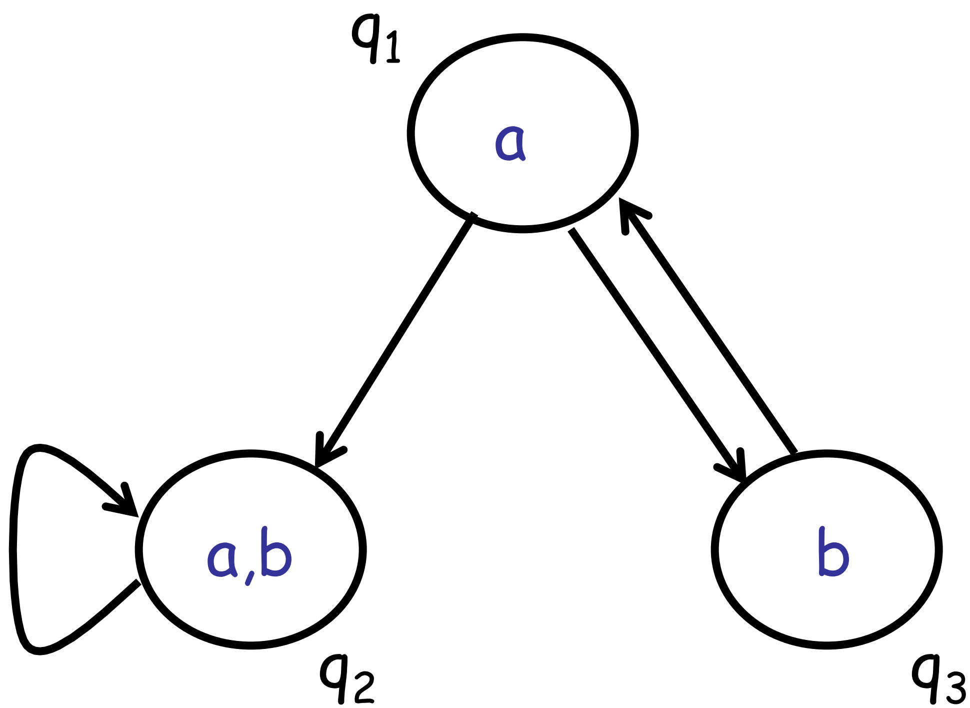
Sets, Strings, Languages, Graphs

- Graph – collection of objects with some (binary) relation among them, G = (N,E), N – nodes, E:= N x N = {(q1,q2), (q1,q3)} - edges



Sets, Strings, Languages, Graphs

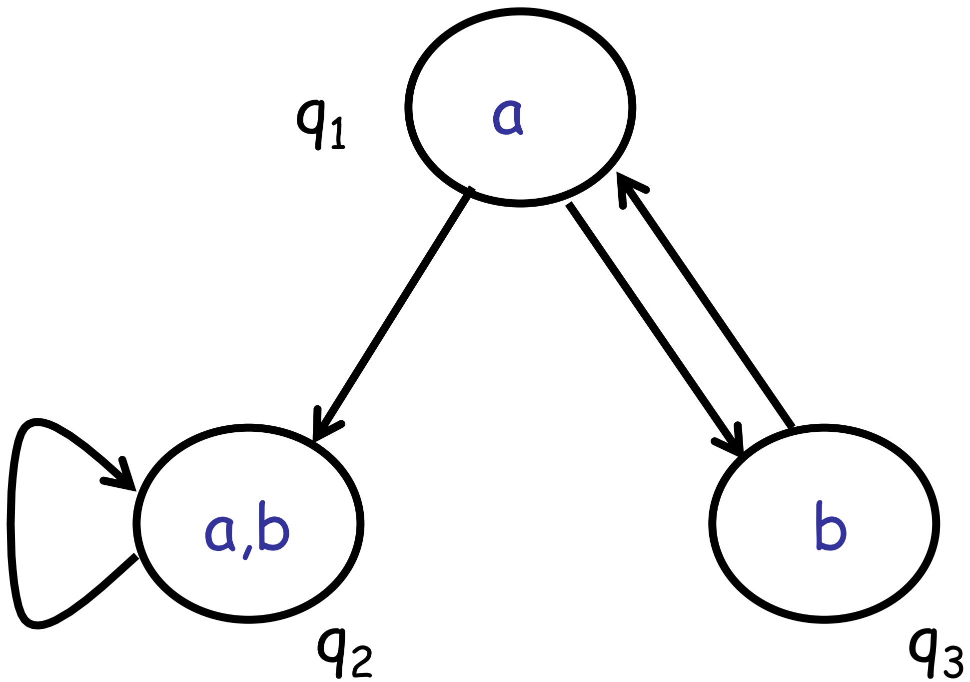
- $L: E \rightarrow D$, D – set of labels
- $G = (N,E,D,L)$, $E := N \times D \times N$



Sets, Strings, Languages, Graphs

- Path – finite (or infinite) sequence of nodes
- Infinite path – at least one node appears infinitely often

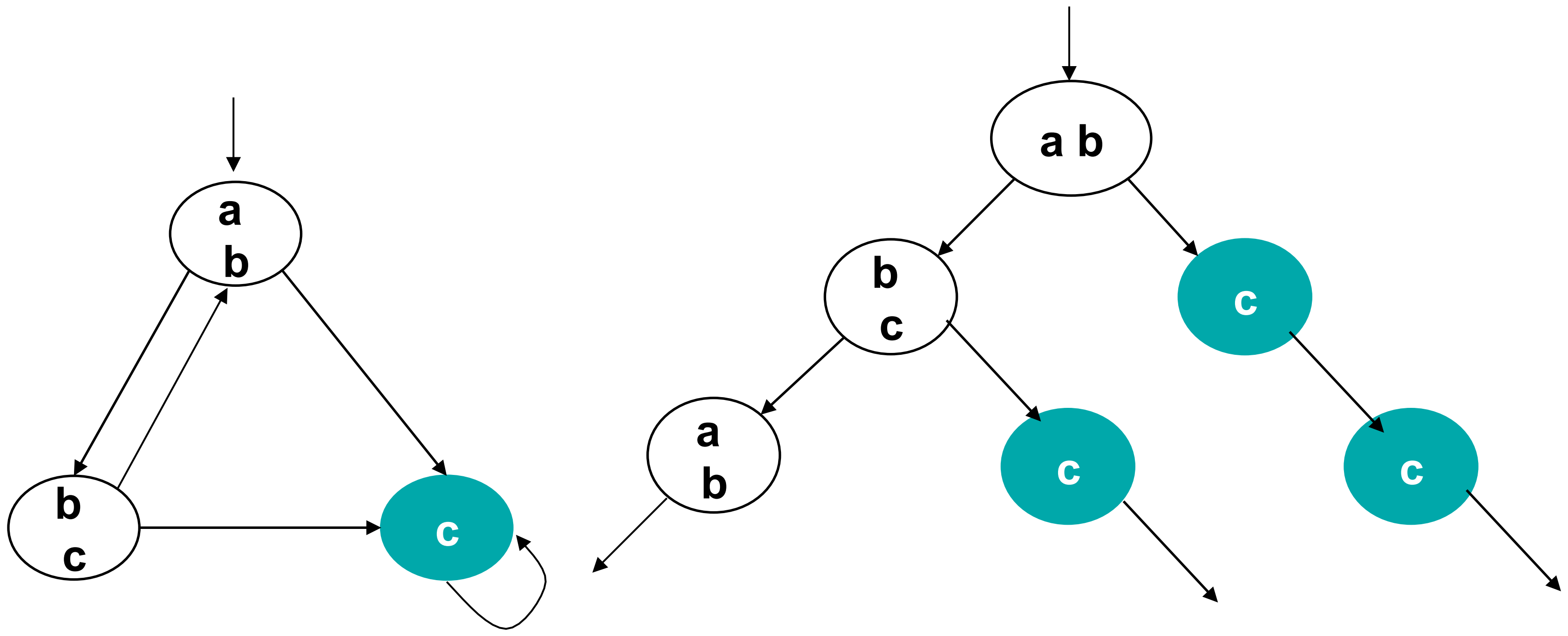
$q_1 \rightarrow q_3 \rightarrow q_1 \rightarrow q_3 \rightarrow q_1 \rightarrow$



Sets, Strings, Languages, Graphs

- A directed graph is a strongly connected graph if ∃ directed path between any pair of its nodes

Sets, Strings, Languages, Graphs



Graph

Infinite Computation Tree

Unwinding (unfolding) of a graph to obtain Infinite Tree

Mathematical Logic

- Mathematical logic formalizes the notion of a *proof*

- Logic Syntax

  How to write legal formulas

- Logic Semantics

  Meaning to each formula

First Order Logic

- Variables range over specific domains (e.g., integers or reals)
- Relations such as ≤, x or +
- Reasons about *all* (∀) the objects in the domain
- Asserts that *exists* (∃) an object satisfying a property

Expression:

term ::= var | const | func(term,…,term)

Assignement:

a : V → D, V –set of Variables, D - domain

First Order Logic Formulas

- simple_form ::= rel(term,…,term) | term $\equiv$ term
- form ::= simple_form | (form $\wedge$ form) | (form $\vee$ form) | ($\neg$ form) | $\forall$var(form) | $\exists$var(form) | true | false

First Order Logic Formulas (2)

- Expressiveness of first order logic (descriptive power of formalism):
  - the graph is undirected
  - there are no unconnected nodes
- Can not express:
  - If graph is finite
  - If it contains cycles
  - If the graph is connected

Propositional Logic

- No quantification, no functions, no relation
- It has a set of propositional variables, *AP* = {true, false}

- form ::= prop | (form $\wedge$ form) | (form $\vee$ form) | ($\neg$ form) | (form $\rightarrow$ form) | true | false