COMP 2049 (AE2LAC) Languages and Computation

Coursework: Automata, Regular Languages, and Context-Free Languages

Spring 2020

Release date: Tuesday, May 5th, 2020

***Deadline*: Wednesday, May 27th, 2020, 16:00**
Cut-off Date: Thursday, May 28th, 2020, 16:00

Total Mark: 100
Weight: 25% of the module mark
How to submit: Via Moodle

1 Finite languages

Task 1.1. *Assume that $L \subseteq \Sigma*$ is a language over the alphabet Σ. Prove that, if L is finite, then L is a regular language.*

Theorem 2: A finite language is regular

Proof: Let us first assume that a language consisting of a single string is regular and prove the theorem by induction. We then prove that a language consisting of a single string is regular.

Claim 1: A language consisting of n strings is regular for any natural number n (that is, a finite language is regular) if { w } is regular for any string w.

Proof of the Claim 1: Proof by induction on the number of strings.

Basis Step: $\emptyset$ (corresponding to n = 0) is a regular language by the Basis Clause of the definition of regular language.

Inductive Step: Assume that a language L consisting of n strings is a regular language (induction hypothesis). Then since { w } is a regular language as proven below, $L \cup \{ w \}$ is a regular language by the definition of regular language.

End of proof of Claim 1

Thus if we can show that { w } is a regular language for any string w, then we have proven the theorem.

Claim 2: Let w be a string over an alphabet Σ. Then { w } is a regular language.

Proof of Claim 2: Proof by induction on strings.

Basis Step: By the Basis Clause of the definition of regular language, $\{\Lambda\}$ and { a } are regular languages for any arbitrary symbol a of Σ .

Inductive Step: Assume that { w } is a regular language for an arbitrary string w over Σ. Then for any symbol a of Σ , { a } is a regular language from the Basis Step. Hence by the Inductive Clause of the definition of regular language { a }{ w } is regular. Hence { aw } is regular.

End of proof for Claim 2

Note that Claim 2 can also be proven by induction on the length of string.

End of proof for Theorem

2 Infinite-state automaton (ISA)

Task 2.1. *Assume that* $\Sigma = \{a, b\}$ *and* $L \subseteq \Sigma*$ *is an* ***arbitrary*** *language over* Σ. *Prove that there exists an ISA* M *such that* $L(M) = L$.

Note: The proof must be clear, complete, and no longer than half a page.

- An arbitrary language has its necessary and sufficient conditions as any subset of {a, b}*
- Assume L is a language that contains single element , it is a regular and finite language, n=1, and we can easily construct an infinite state automaton, which is an infinite collection of finite automaton.
- Now that we have M elements in the language, for you have given a finite number , this is a finite language. In the previous section task 1.1 we have proven that finite language is regular. And thus we could prove that there is a finite automaton to accept the language. And even if you couldn't do so, you could construct M automaton together and let them connected together. Such union is still a finite automaton. The infinite automaton is a finite automaton adding infinite states and infinite final states. Thus we have proven that when n=M, an infinite automaton definitely can be made.
- For the rest of the number M+1 ,M+2…M+INFINITE , from deductive reasoning, we could always add more states and final states to let the automaton terminated such that L(M)=L
- Supplement proof for those language which is finite or regular. Then from the given information "if L is a regular language, then there exists an ISA that accepts L."

3 Context-free languages (CFLs)

For each $n \geq 1$, let $b(n)$ denote the binary representation of n, and let $\text{revb}(n)$ denote the reverse of $b(n)$. Some examples of these strings are provided for various values of n in the following table:

n **in decimal**	5	6	12	13	43	44
$b(n)$	101	110	1100	1101	101011	101100
$\text{revb}(n)$	101	011	0011	1011	110101	001101

Consider the alphabet $\Sigma = \{0, 1, .\}$, and define the language $L_r \subseteq \Sigma*$ as follows:

$$L_r := \{\text{revb}(n)\ .\ b(\bar{n} + 1) \mid n \geq 1\}$$

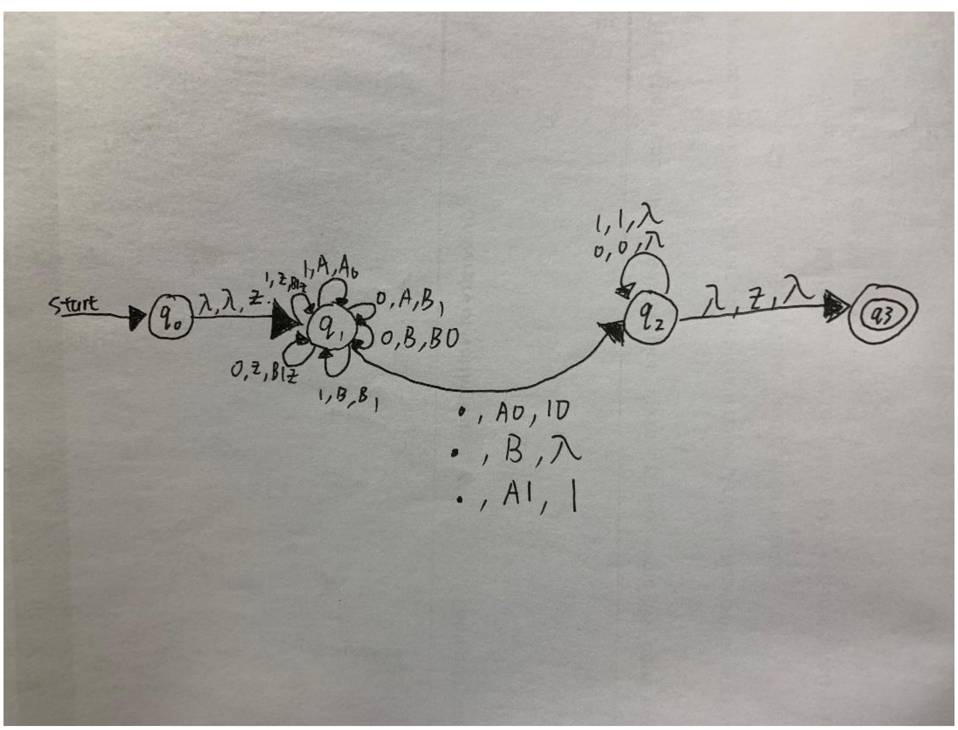
$$L_r = \{1\ .\ 10, 01\ .\ 11, 11\ .\ 100, 001\ .\ 101, \cdots\}$$

Task 3.1. *Show that L_r is a context-free language by constructing a pushdown automaton (PDA) that accepts L_r. In particular, you are required to:*

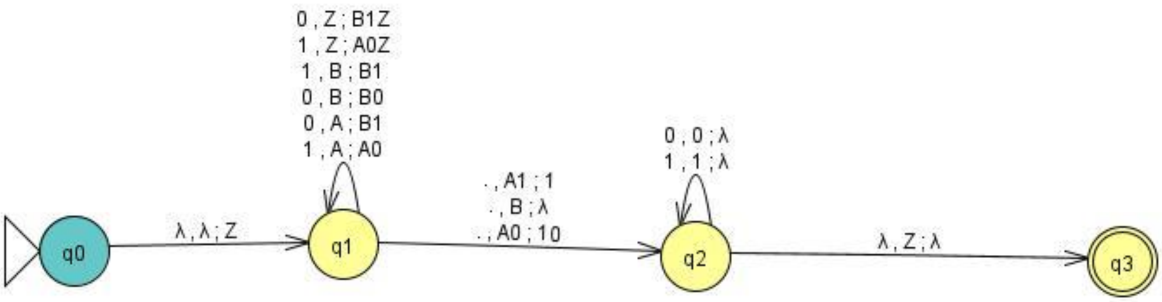
(a) Provide a clear description of the main idea of the design of the PDA and what the stack is used for.

So this PDA is used to determine if this input string is legal for acceptance, or it will be rejected. After reaching the end of string if the automaton is entering the final state then it is belonging to L_r , if it doesn't entered the final state it is rejected. Stack is used to store what is read in and help with the transition.
The thing you have to notice is that this b(n) + 1=b(n+1), so from right to left you have to find the first zero , all the flip(carry) will stop here. First read in all the numbers in the input, and after the dot, begin to check if every input is the same ,after first zero is reached check if every number is flipped.

(b) Draw a transition diagram for the constructed PDA.



(c) Implement the PDA in JFLAP.



Task 3.2. *Design a context free grammar G for which $L_r = L(G)$. Specifically, you are required to:*

(a) Write down all the production rules clearly.

$S \rightarrow SS$

$S \rightarrow 1S$

$S \rightarrow 0S$

$S \rightarrow .S$

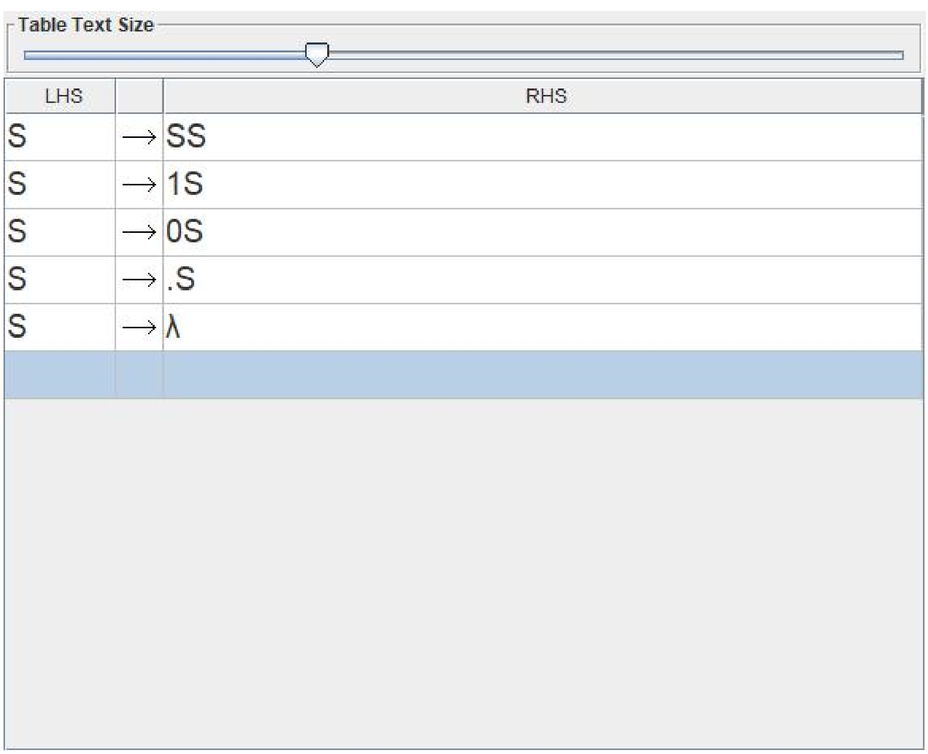
$S \rightarrow \lambda$

(b) Give a verbal account of the main points of your answer, such as, what each variable is used for. This should not take more than a paragraph.

$G = (S, \{0,1,.\}, R, S)$

A context-free grammar G is defined by the 4-tuple $G = (V, \sum, R, S)$
The first parameter V is a finite set which contains a lot of variables , in this case , we just take single variable S
The second parameter is sigma which is all the possible alphabet of the input string
The members of R are called the (rewrite) rules or productions of the grammar. (also commonly symbolized by a P)
The last parameter S is the start variable (or start symbol), used to represent the whole sentence (or program). It must be an element of V. Here it is just S.

(c) Implement the grammar in JFLAP.



LHS		RHS
S	→	SS
S	→	1S
S	→	0S
S	→	.S
S	→	λ

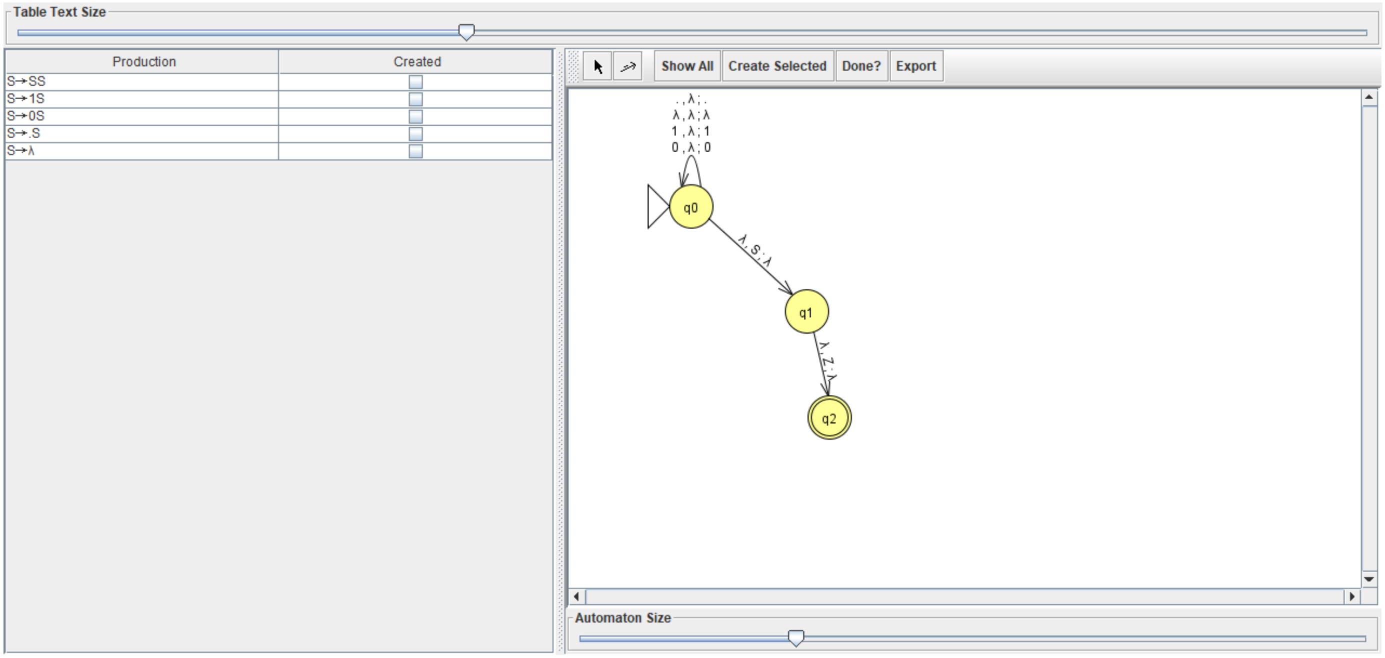


Table Text Size
Production
Created
S→SS
S→1S
S→0S
S→.S
S→λ
Show All
Create Selected
Done?
Export
. , λ ; .
λ , λ ; λ
1 , λ ; 1
0 , λ ; 0
q0
λ , S ; λ
q1
λ , Z ; λ
q2
Automaton Size

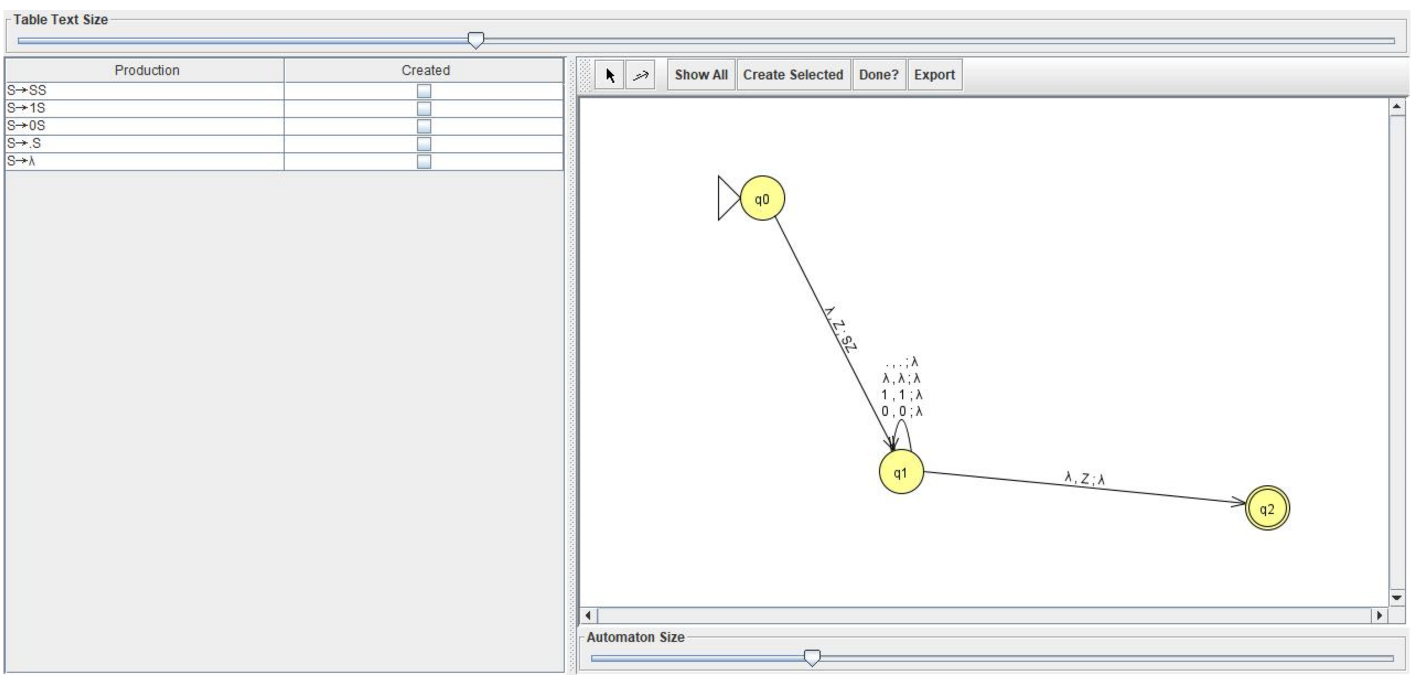


Table Text Size
Production
Created
S→SS
S→1S
S→0S
S→.S
S→λ
Show All
Create Selected
Done?
Export
q0
λ , Z ; SZ
. , . ; λ
λ , λ ; λ
1 , 1 ; λ
0 , 0 ; λ
q1
λ , Z ; λ
q2
Automaton Size