

9/17(木) 제4강 FA의 종류 (DFA의 확장)

이 부분은
항상 기억한다
proof

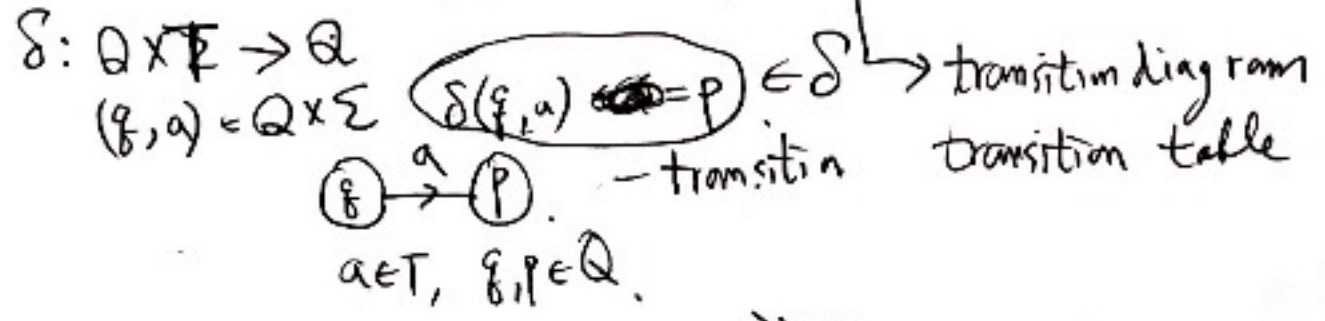
1 Automata: The Methods and the Madness	1
1.1 Why Study Automata Theory?	2
1.1.1 Introduction to Finite Automata	2
1.1.2 Structural Representations	4
1.1.3 Automata and Complexity	5
1.2 Introduction to Formal Proof	5
1.2.1 Deductive Proofs	6
1.2.2 Reduction to Definitions	8
1.2.3 Other Theorem Forms	10
1.2.4 Theorems That Appear Not to Be If-Then Statements	13
1.3 Additional Forms of Proof	13
1.3.1 Proving Equivalences About Sets	14
1.3.2 The Contrapositive	14
1.3.3 Proof by Contradiction	16
1.3.4 Counterexamples	17
1.4 Inductive Proofs	19
1.4.1 Inductions on Integers	19
1.4.2 More General Forms of Integer Inductions	22
1.4.3 Structural Inductions	23
1.4.4 Mutual Inductions	26
1.5 The Central Concepts of Automata Theory	28
1.5.1 Alphabets (Vocabulary)	28
1.5.2 Strings	29
1.5.3 Languages	30
1.5.4 Problems	31
1.6 Summary of Chapter 1	33
1.7 Gradiance Problems for Chapter 1	35
1.8 References for Chapter 1	36
2 Finite Automata	37
2.1 An Informal Picture of Finite Automata	38
2.1.1 The Ground Rules	38
2.1.2 The Protocol	39
2.1.3 Enabling the Automata to Ignore Actions	41
2.1.4 The Entire System as an Automaton	43
2.1.5 Using the Product Automaton to Validate the Protocol	44
2.2 Deterministic Finite Automata (DFA)	45
2.2.1 Definition of a Deterministic Finite Automaton	45
2.2.2 How a DFA Processes Strings	46
2.2.3 Simpler Notations for DFA's (from Symbol to)	47
2.2.4 Extending the Transition Function to Strings	49
2.2.5 The Language of a DFA	52
2.2.6 Exercises for Section 2.2	52
2.3 Nondeterministic Finite Automata (NFA)	55
2.3.1 An Informal View of Nondeterministic Finite Automata	55
2.3.2 Definition of Nondeterministic Finite Automata	57
2.3.3 The Extended Transition Function	58
2.3.4 The Language of an NFA	59
2.3.5 Equivalence of Deterministic and Nondeterministic Finite Automata	60
2.3.6 A Bad Case for the Subset Construction	64
2.3.7 Exercises for Section 2.3	65
2.4 An Application: Text Search (Read text)	68
2.4.1 Finding Strings in Text	68
2.4.2 Nondeterministic Finite Automata for Text Search	69
2.4.3 A DFA to Recognize a Set of Keywords	70
2.4.4 Exercises for Section 2.4	71
2.5 Finite Automata With Epsilon-Transitions (ϵ -NFA)	72
2.5.1 Uses of ϵ -Transitions	72
2.5.2 The Formal Notation for an ϵ -NFA	73
2.5.3 Epsilon-Closures	74
2.5.4 Extended Transitions and Languages for ϵ -NFA's	75
2.5.5 Eliminating ϵ -Transitions	77
2.5.6 Exercises for Section 2.5	79
2.6 Summary of Chapter 2	80
2.7 Gradiance Problems for Chapter 2	80
2.8 References for Chapter 2	83

2.2 Deterministic Finite Automata (상어) 기호는 보트인의 명도준

2.2.1 Definition of DFA

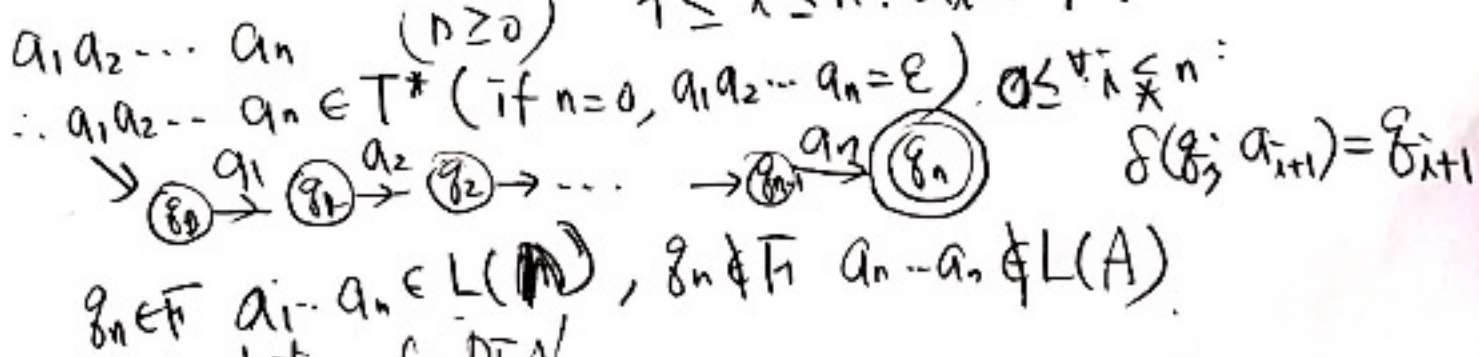
$$A = (Q, T, \delta, q_0, F)$$

- (1) $Q \dots$ a (finite) set of states, $q \in Q$
- (2) $T \dots$ a " " of terminal symbols, $a \in T$
- (3) $\delta \dots$ a " " of transition functions δ

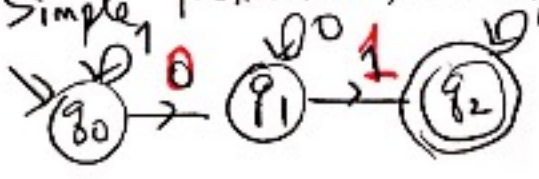


- (4) $q_0 \in Q \dots$ called an initial state $\rightarrow \odot$
- (5) $F \subseteq Q \dots$ a (finite) set of final states. \odot
 non-empty,

2.2.2 How a DFA Processes strings $1 \leq i \leq n: a_i \in T$

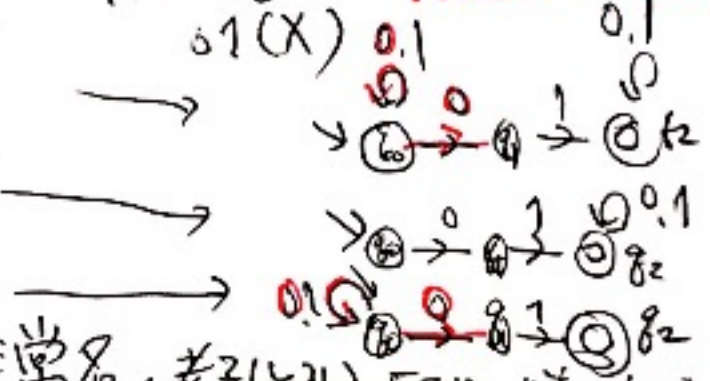


2.2.3 Simple notation for DFA's



01 substring의 처리는 스트림
 $01(X) \dots$ Nondeterministic FA

Substring $(011)^*$
 Prefix $01(011)^*$
 Postfix $(011)^* 01$



學而之名錄 vs 名可名 非常名. 老子(노자) 도덕경 (道德經)

2.3 Nondeterministic Finite Automata (NFA)

$$N = (Q, T, \delta, q_0, F)$$

$Q, T, q_0, F \equiv$ DFA와 같음. 다만

$$\delta: Q \times \Sigma \rightarrow 2^Q$$

$$\delta(q, a) = \{p_1, p_2, \dots, p_k\}$$



DFA는 NFA의 일종이다.

중 destination state를 하나 뿐인 경우이다.

모든 NFA는 DFA로 바꿀 수 있다! - subset construction

