**Compiler**

**Team-project #1**

**-Lexical Analyzer-**



Team 52

20161851 조민수

**Table of Contents**

1. Defining tokens

2. Regular expressions & NFA & DFA

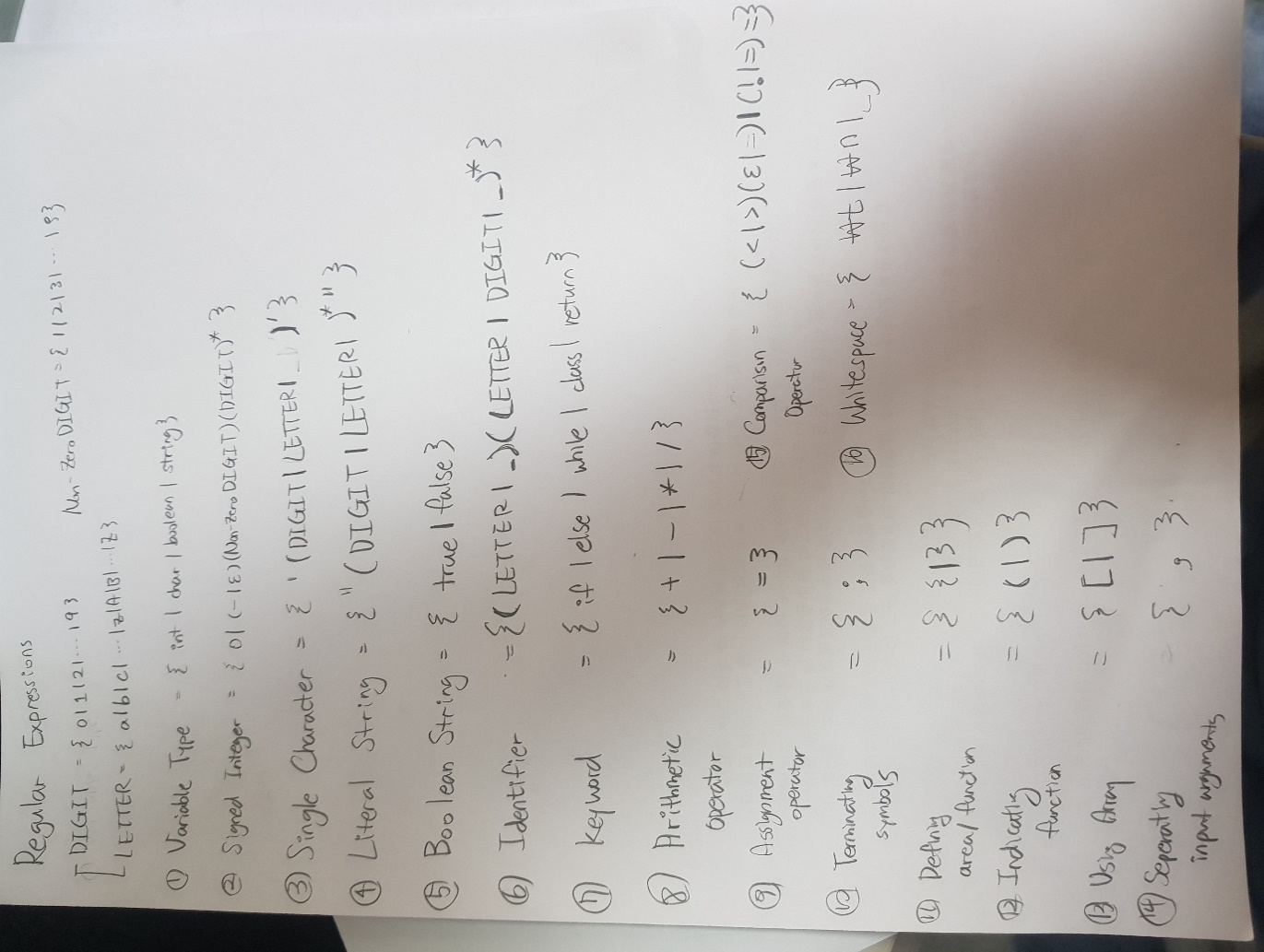
3. Implementation

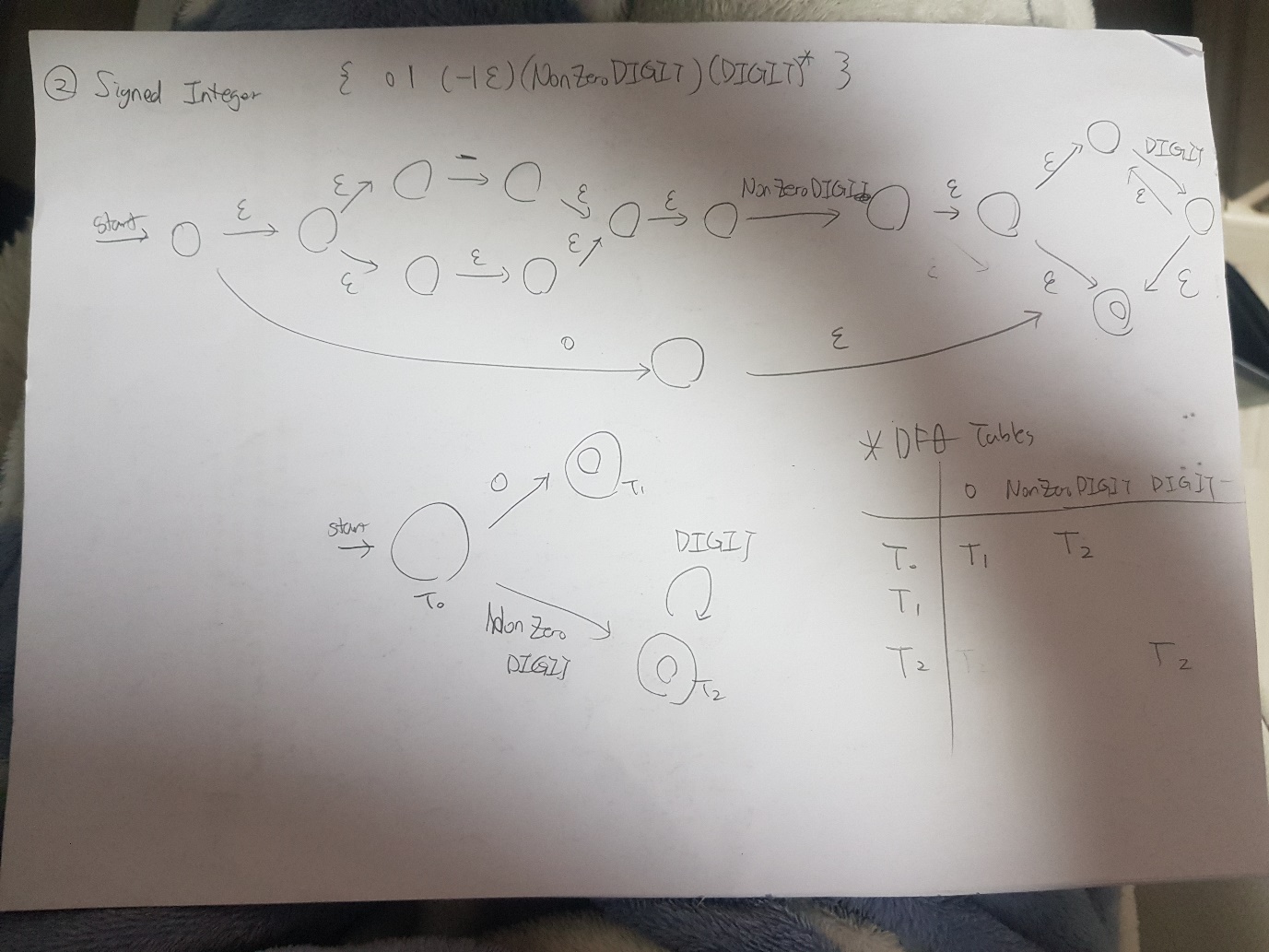
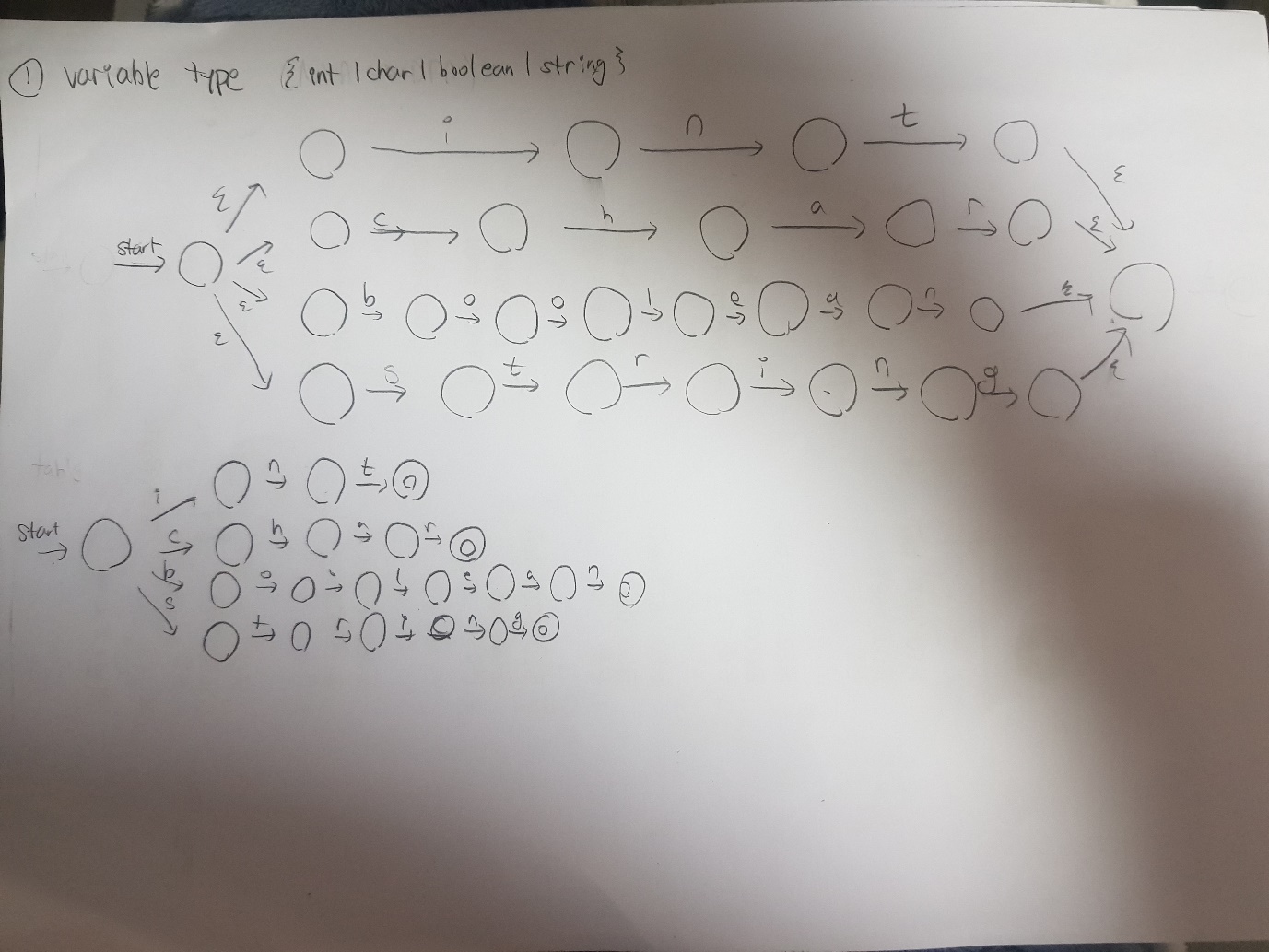
4. Details

**1. Defining Tokens**

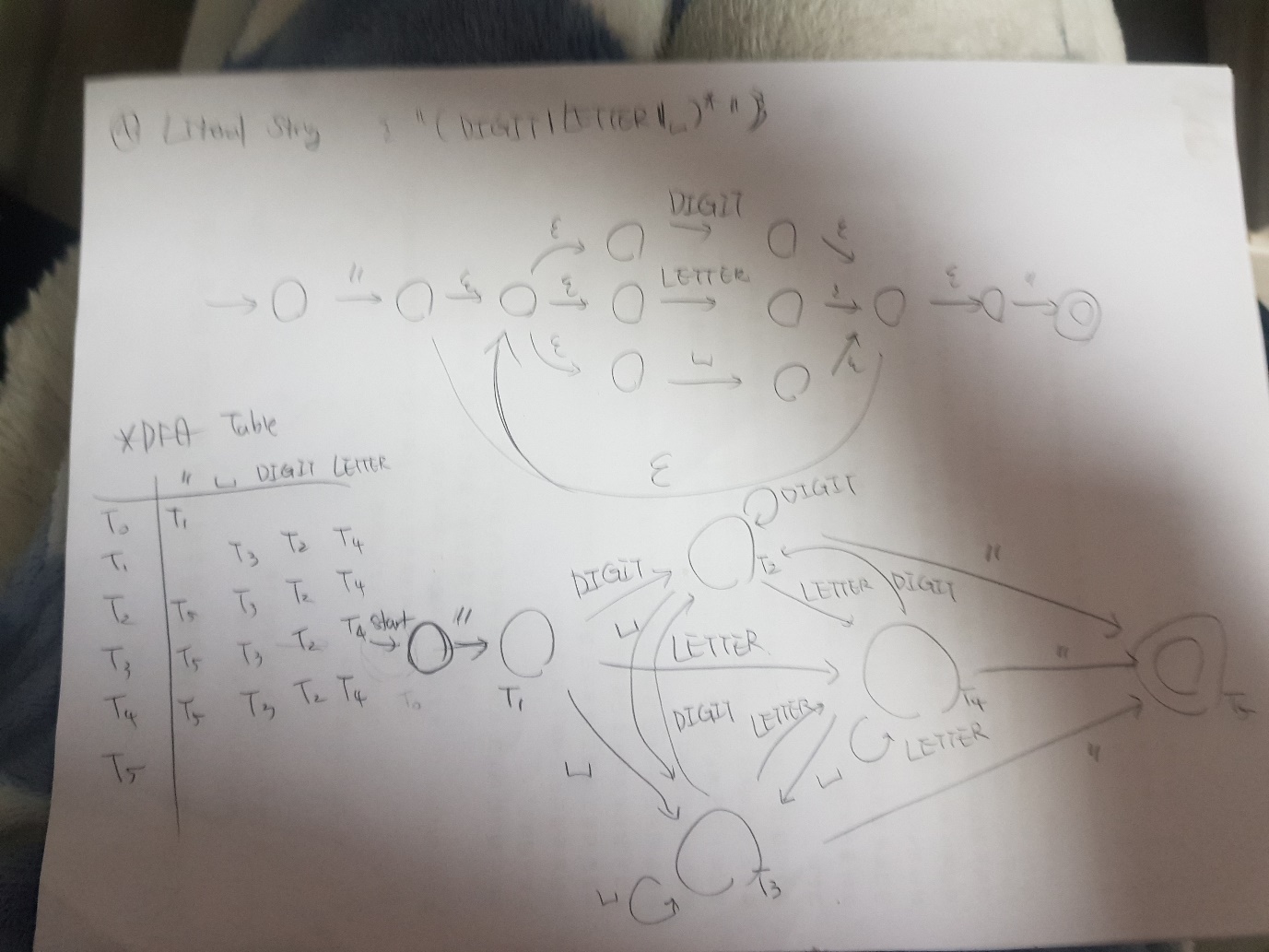
|  |  |  |
| --- | --- | --- |
| Lists | lexemes | Token |
| Variable type | int | <VTYPE, int> |
| char | <VTYPE, char> |
| boolean | <VTYPE, bool> |
| String | <VTYPE, string> |
| Signed Integer | 0, 1, 12312 | <INTEGER,N> |
| Single Character | ‘a’, ‘1‘, ‘’ | <char,N> |
| Boolean String | True | TRUE |
| False | FALSE |
| Literal String | ¡°My name is minsu¡± | <string,N> |
| Identifier | input, count | <ID,N> |
| Keywords | if | IF |
| else | ELSE |
| while | WHILE |
| class | CLASS |
| return | RETURN |
| Arithmetic Operators | + | <OP,+> |
| - | <OP,-> |
| \* | <OP,\*> |
| / | <OP,/> |
| Assignment Operator | = | <ASSIGN> |
| Comparison Operators  Comparison  operators | > | <COMPARISON, >> |
| < | <COMPARISON, <> |
| == | <COMPARISON, ==> |
| != | <COMPARISON, !=> |
| >= | <COMPARISON, >=> |
| <= | <COMPARISON, <=> |
| Terminating Symbol of Statements | ; | <SEMI> |
| A pair of symbols for defining  area/ scope of variables and functions | { | LBRACE |
| } | RBRACE |
| A pair of symbols for indicating  a function / statement | ( | LPAREN |
| ) | RPAREN |
| A pair of symbols for using array | [ | LSBRACKET |
| ] | RSBRACKET |
| A symbol for separating  input arguments in functions | , | <COMMA> |
| Whitespaces | \t | IGNORED |
| \n | IGNORED |
|  | IGNORED |

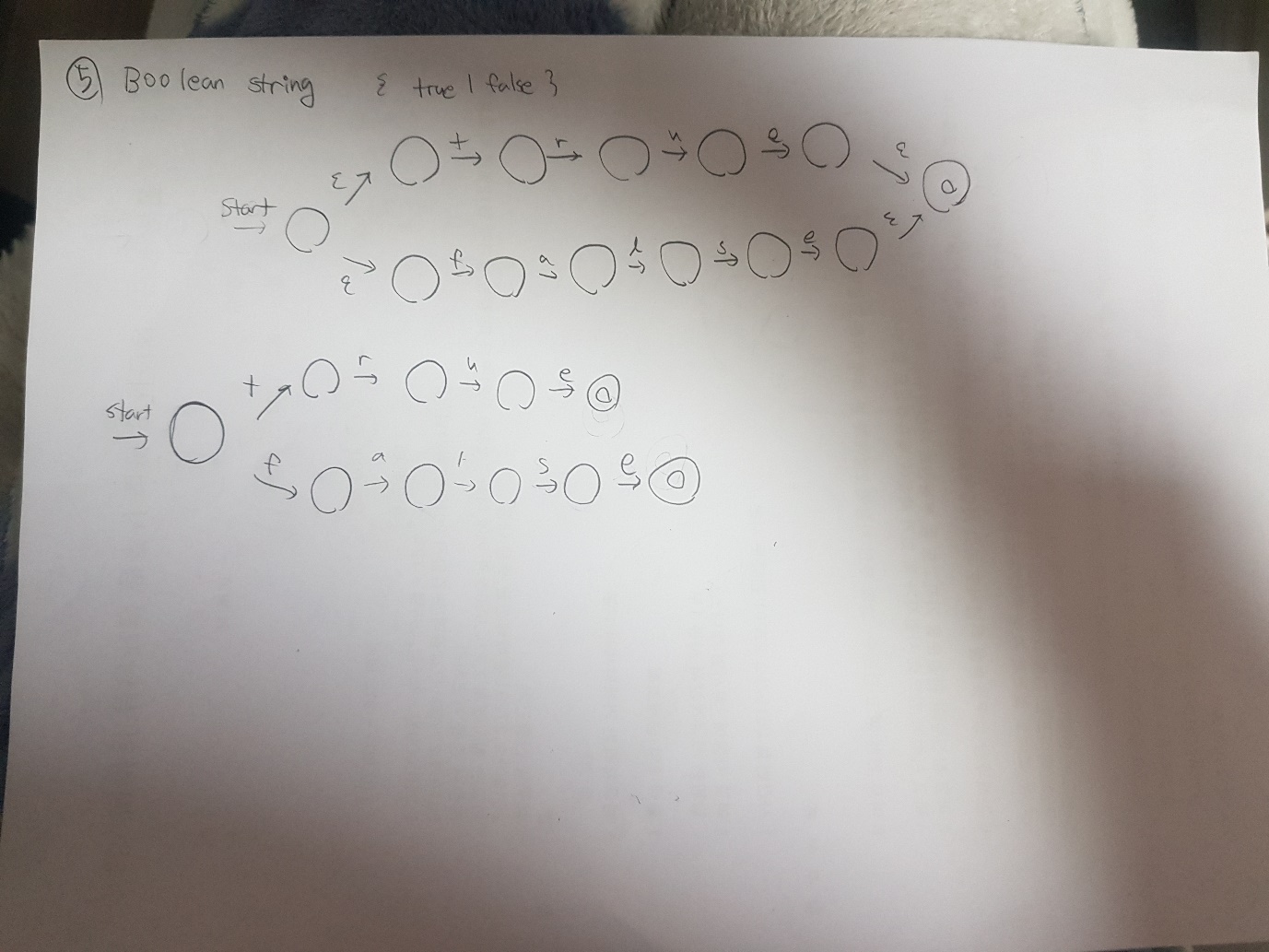
**2. Regular expressions & NFA & DFA**

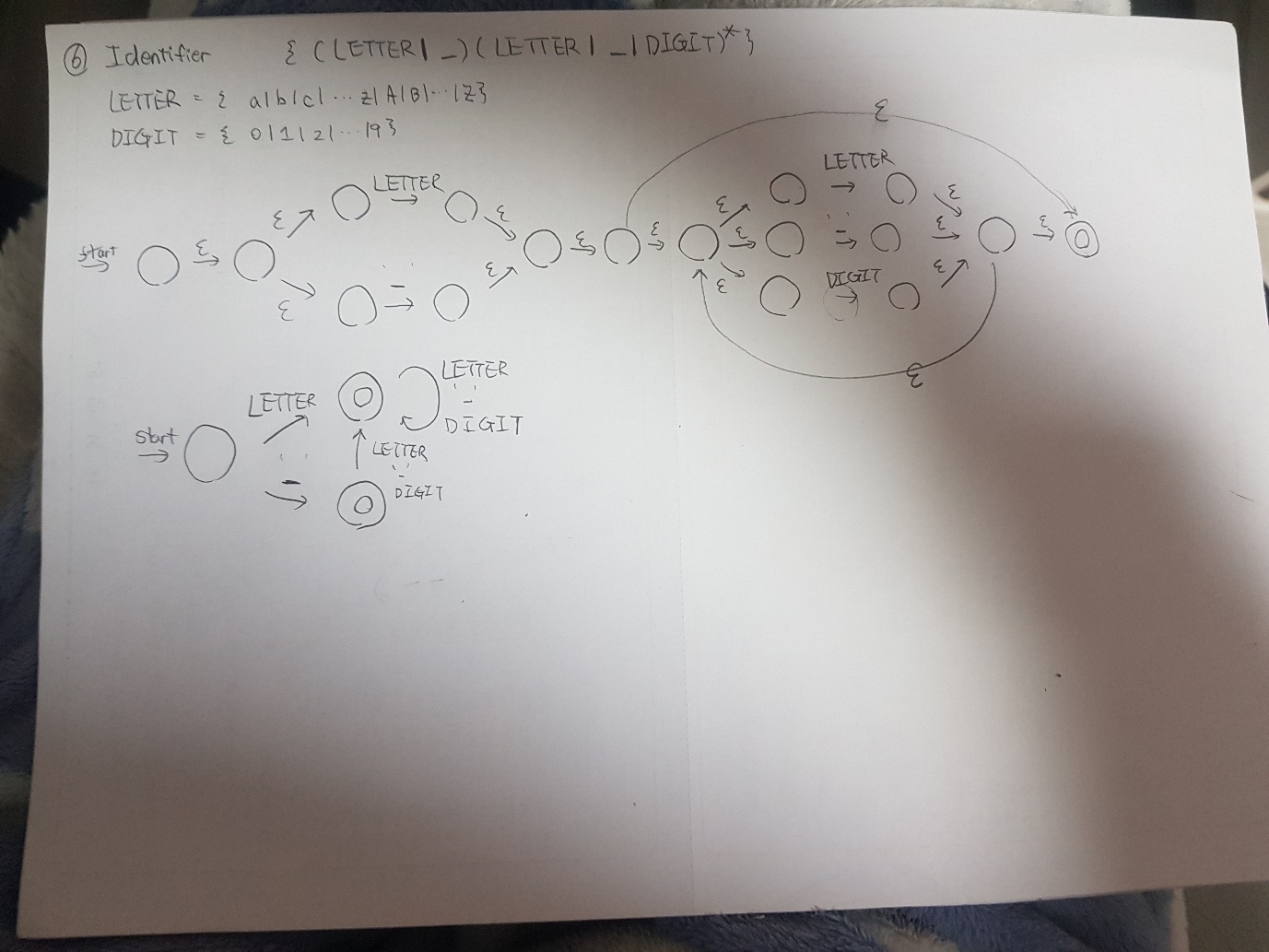
****

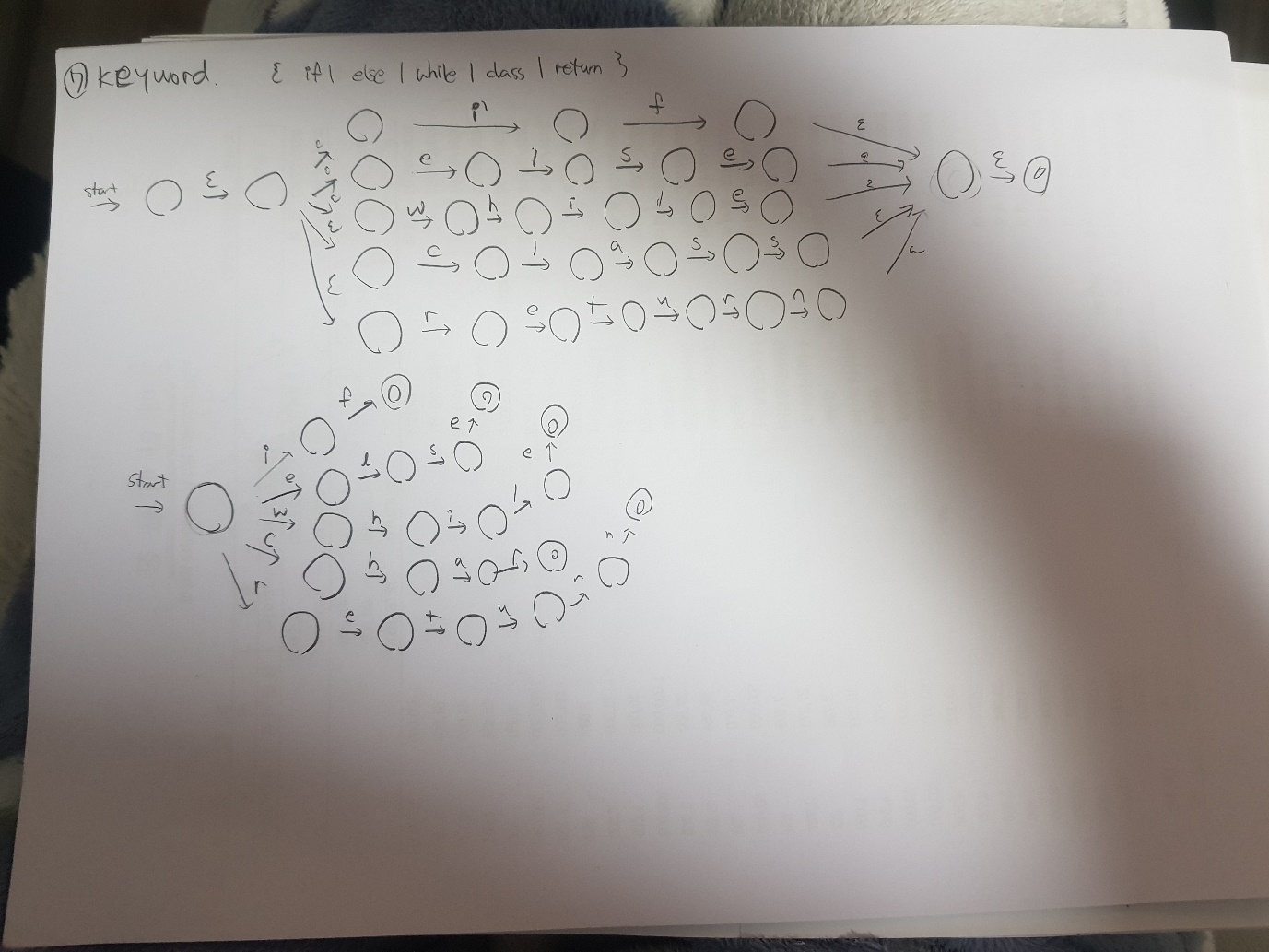
****

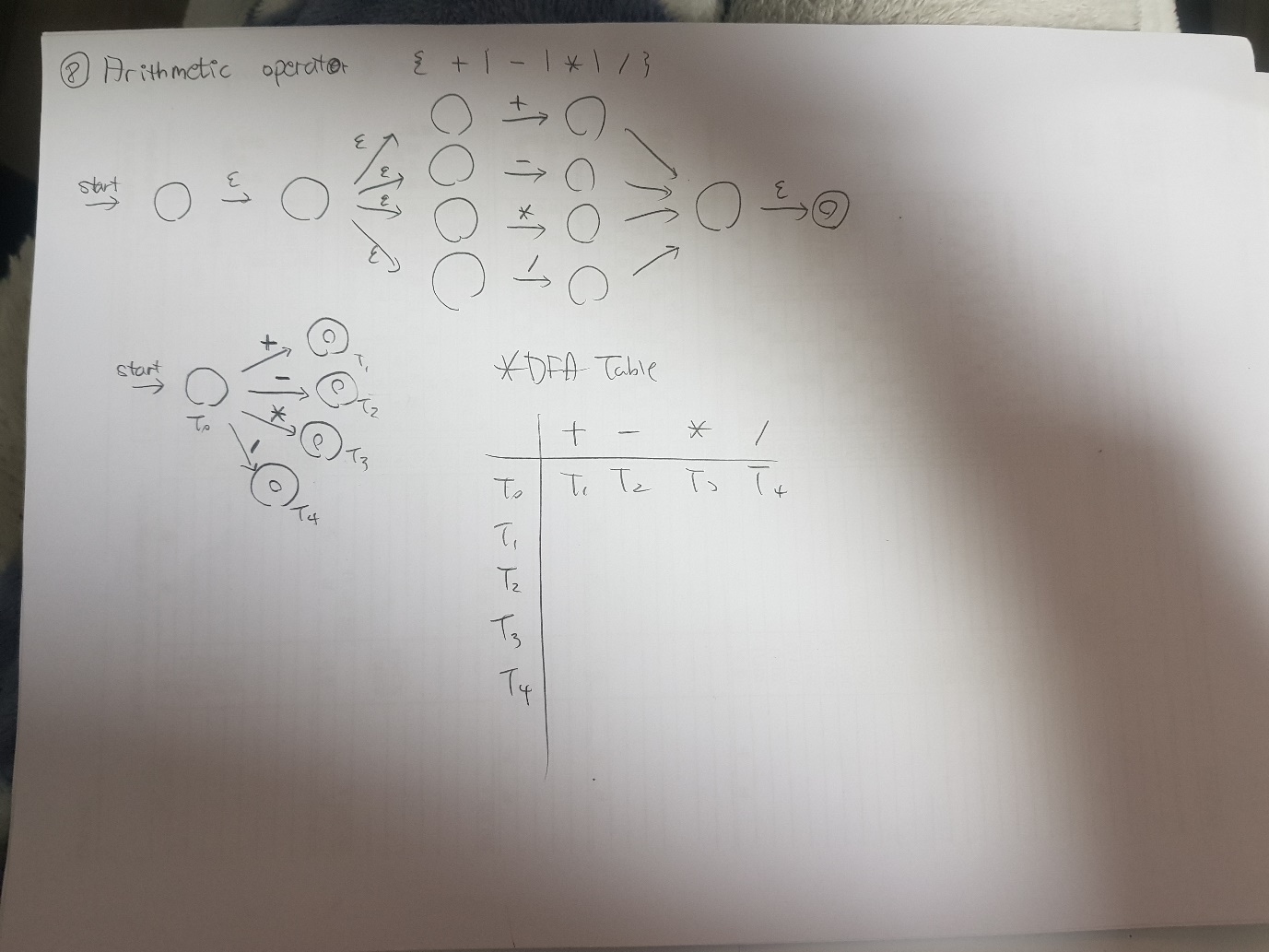
****

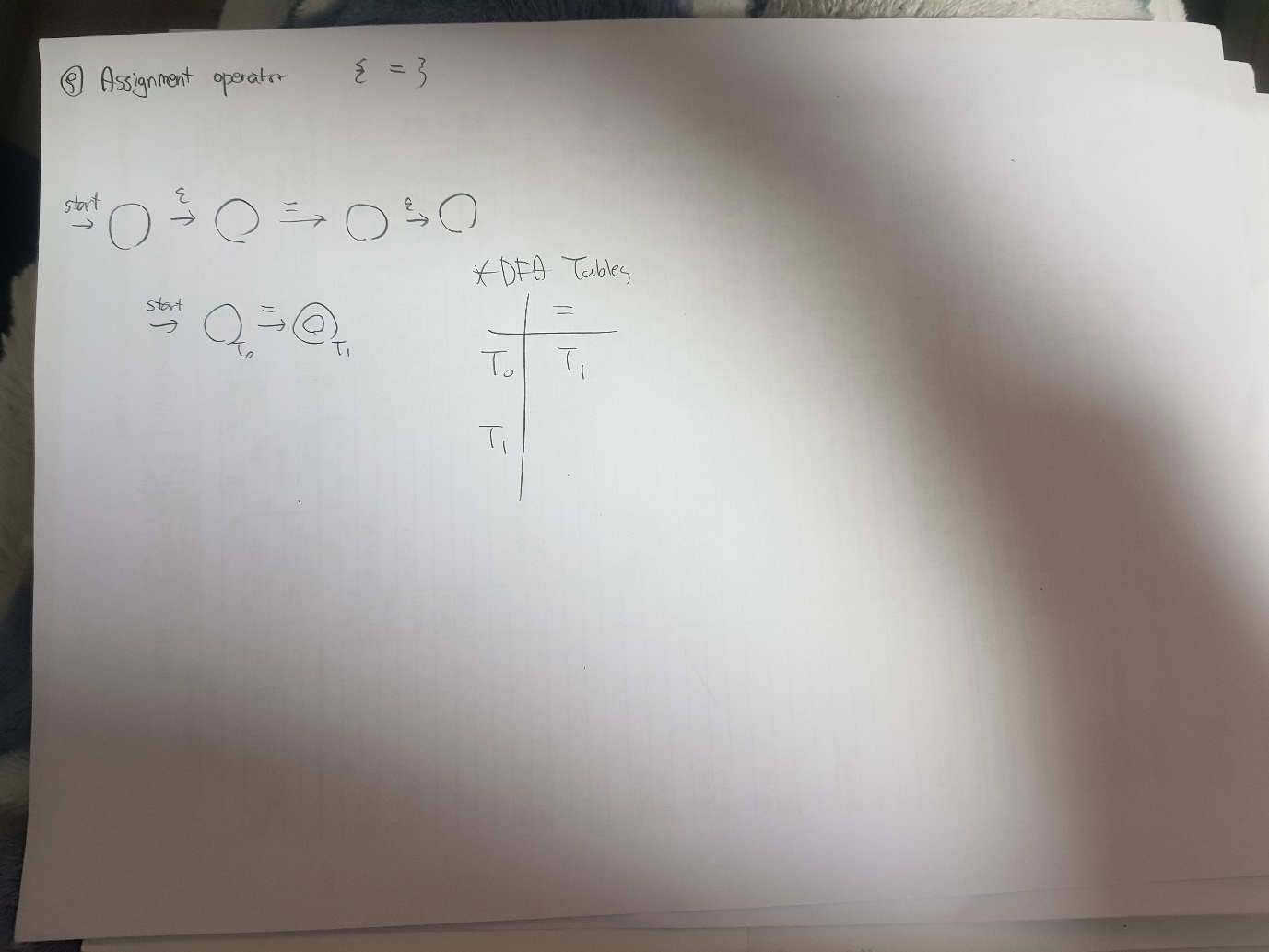
****

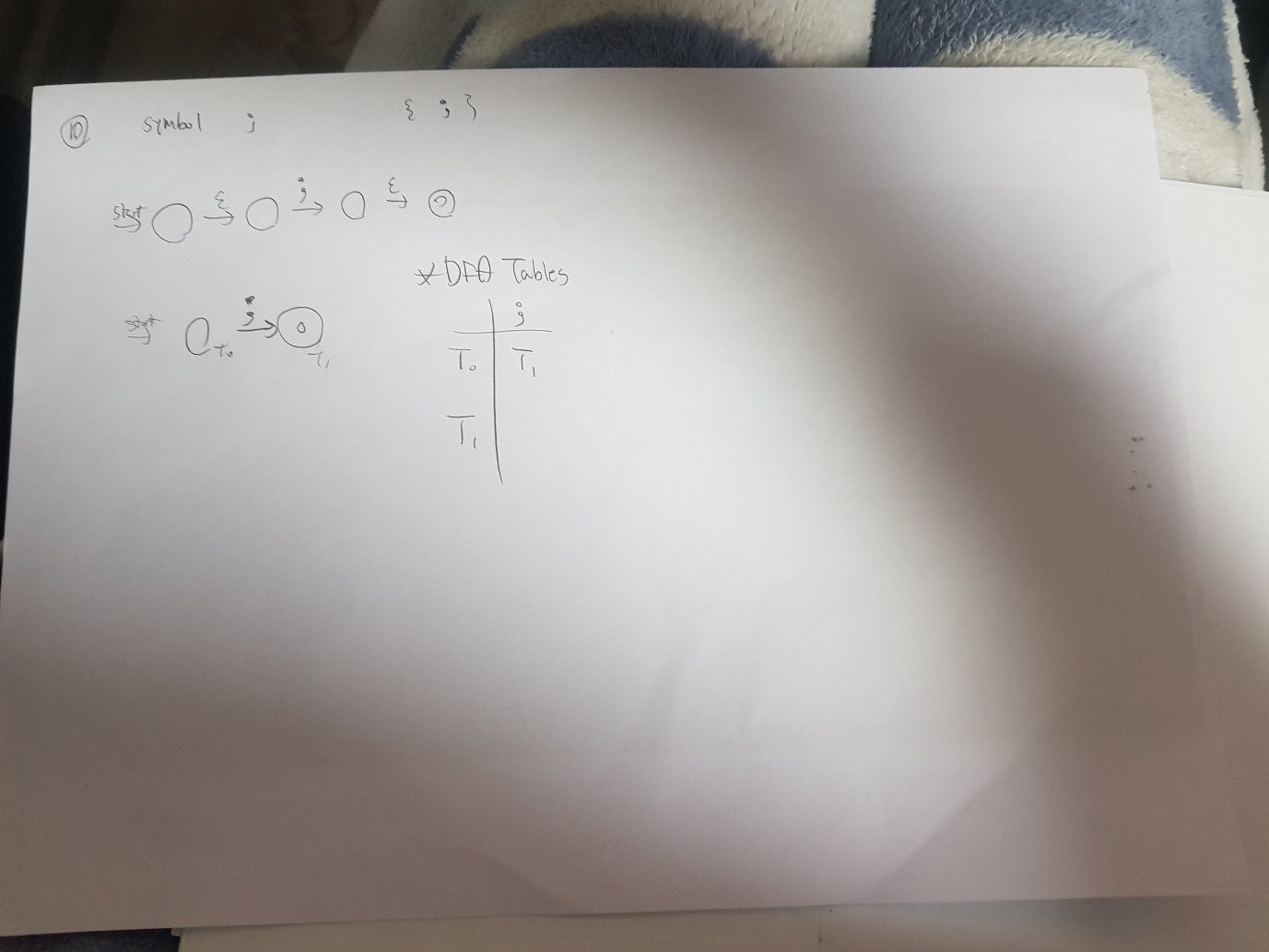
****

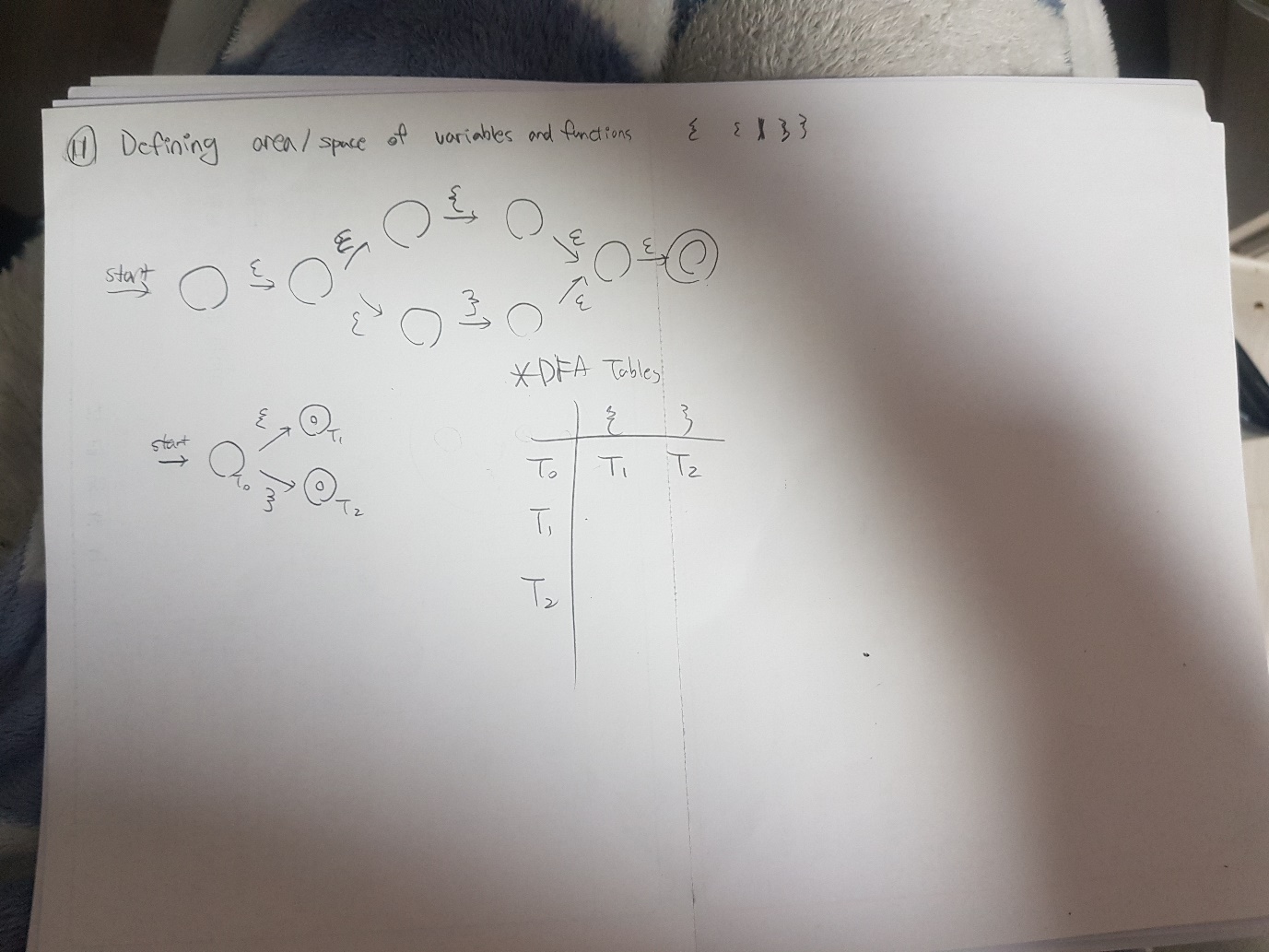
****

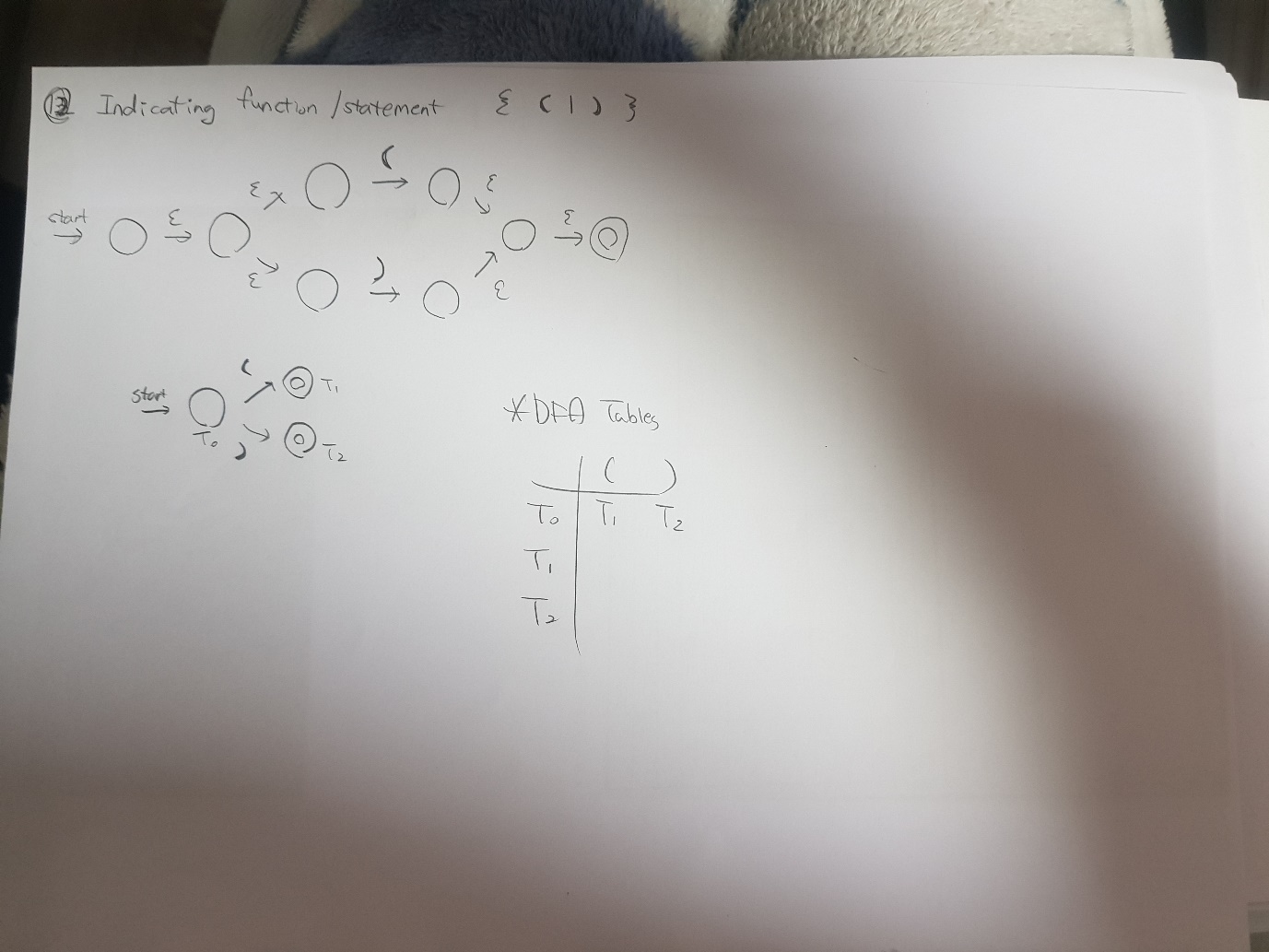
****

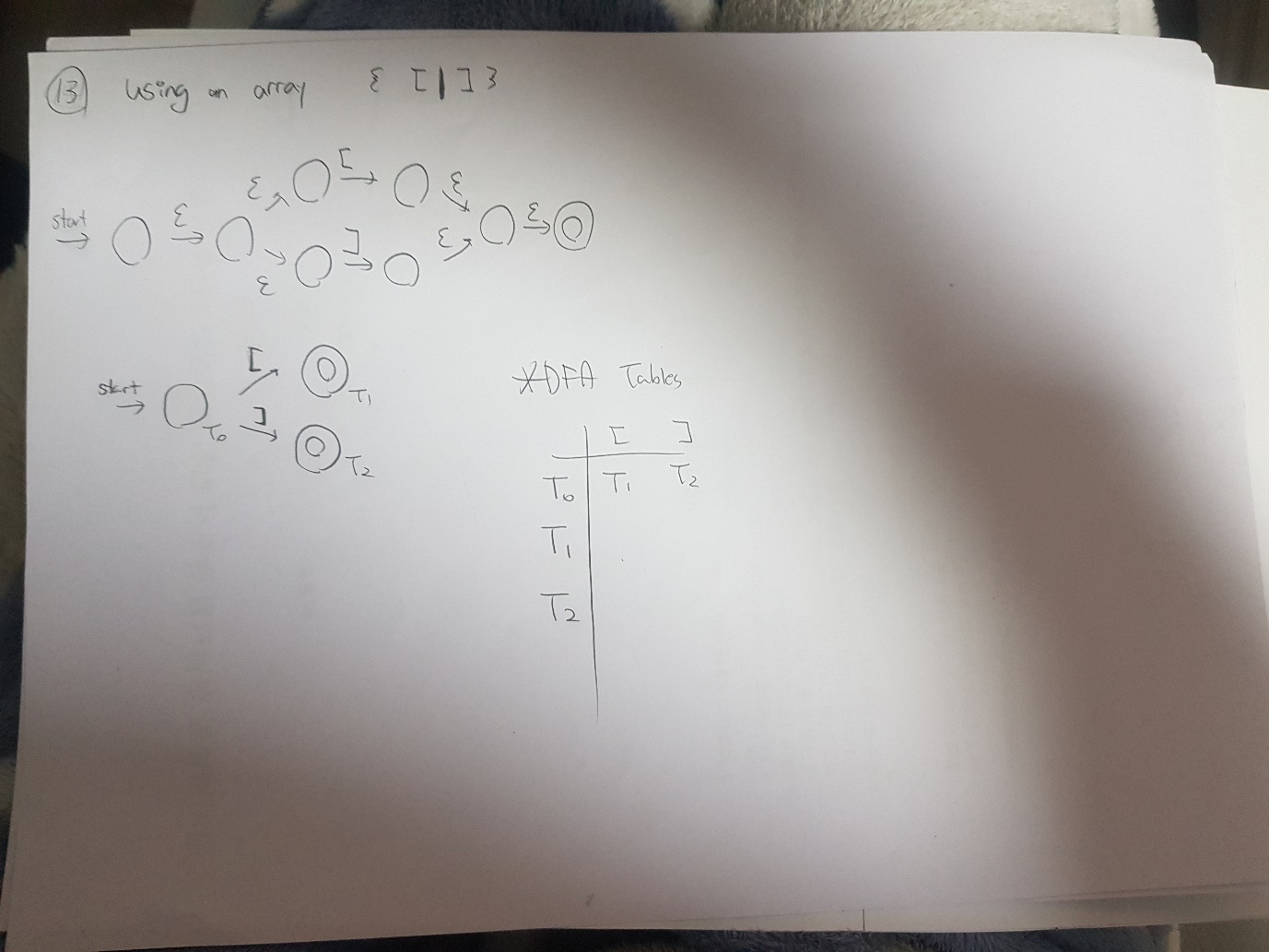
****

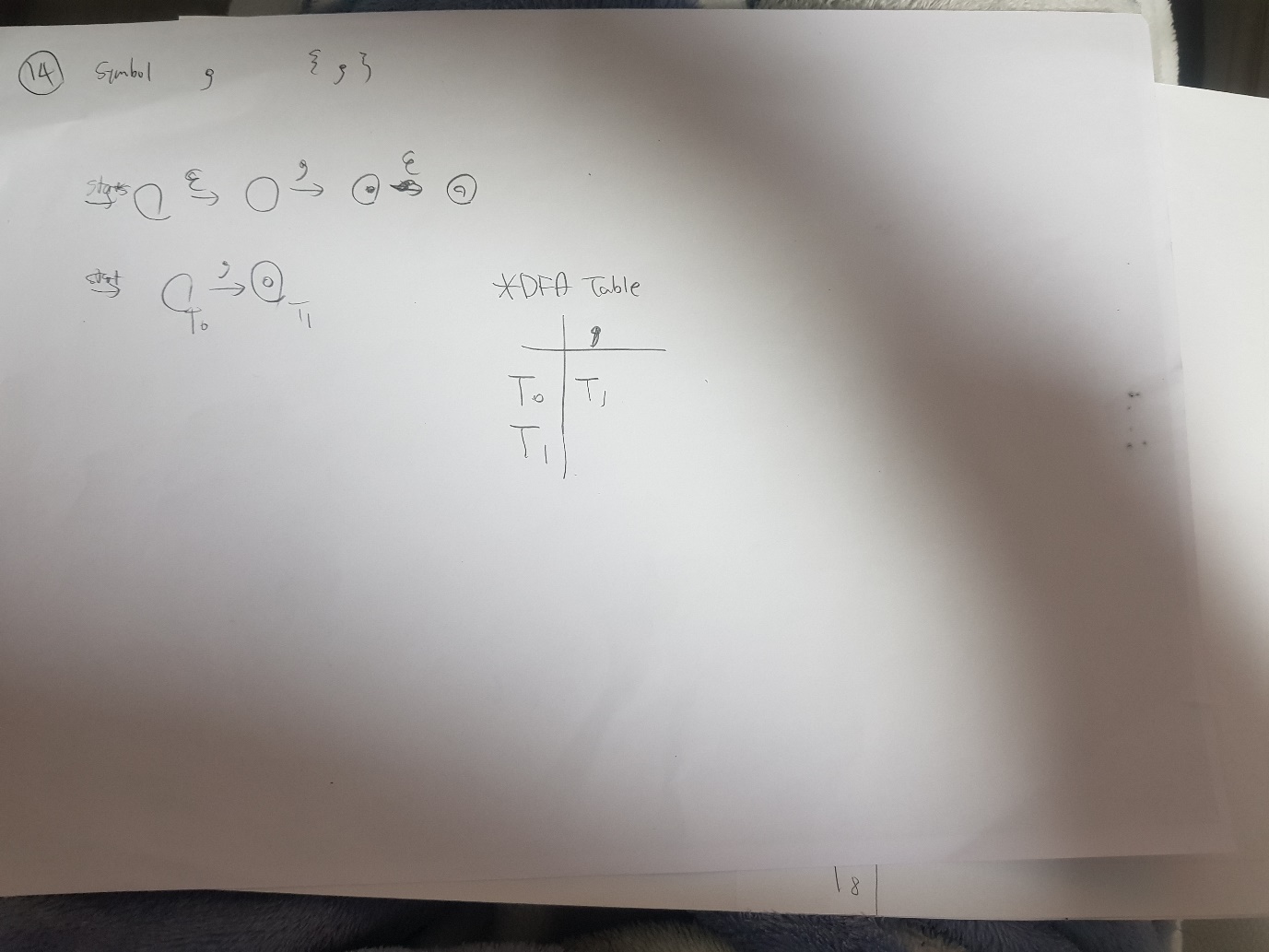
****

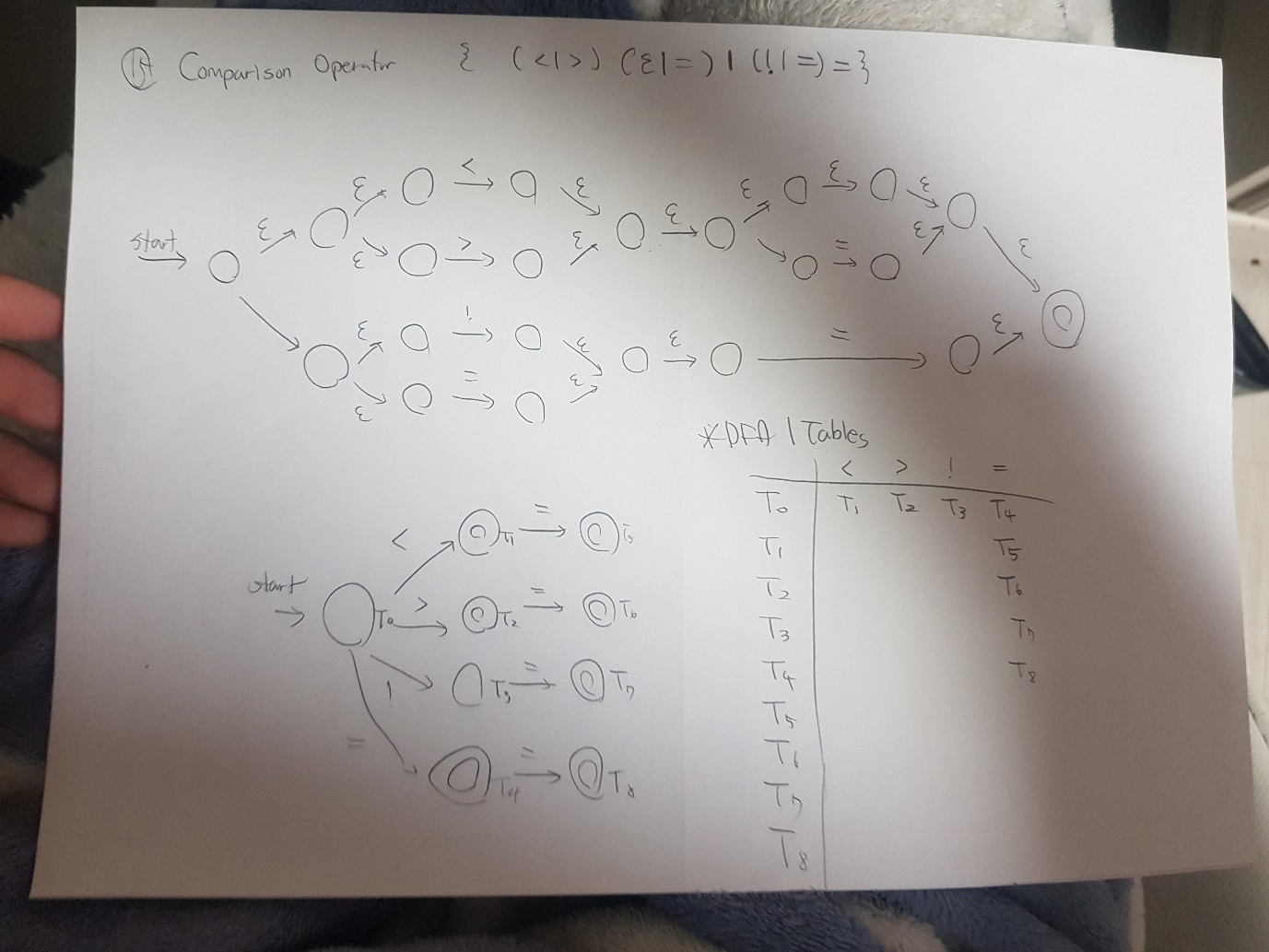
****

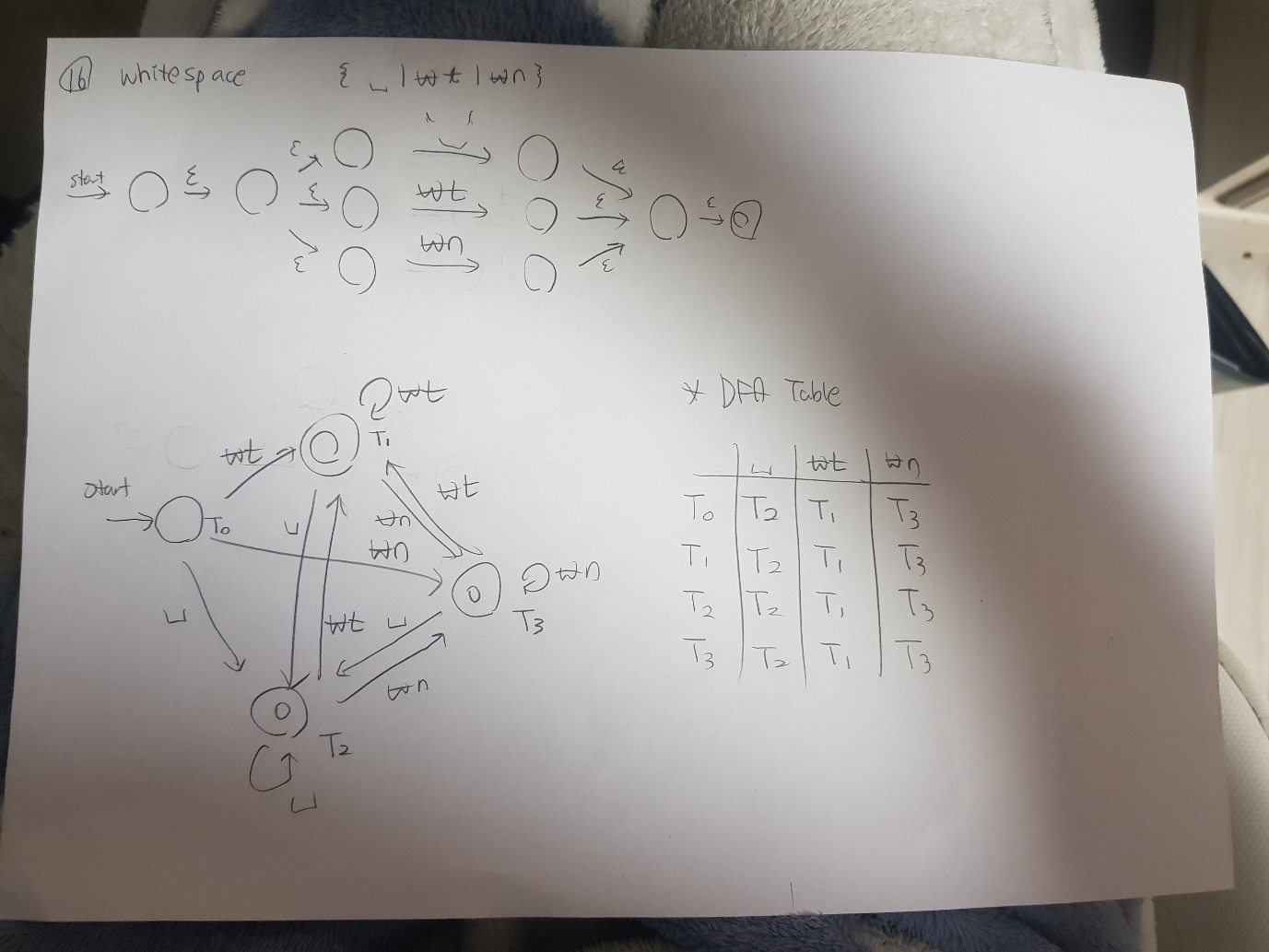
****

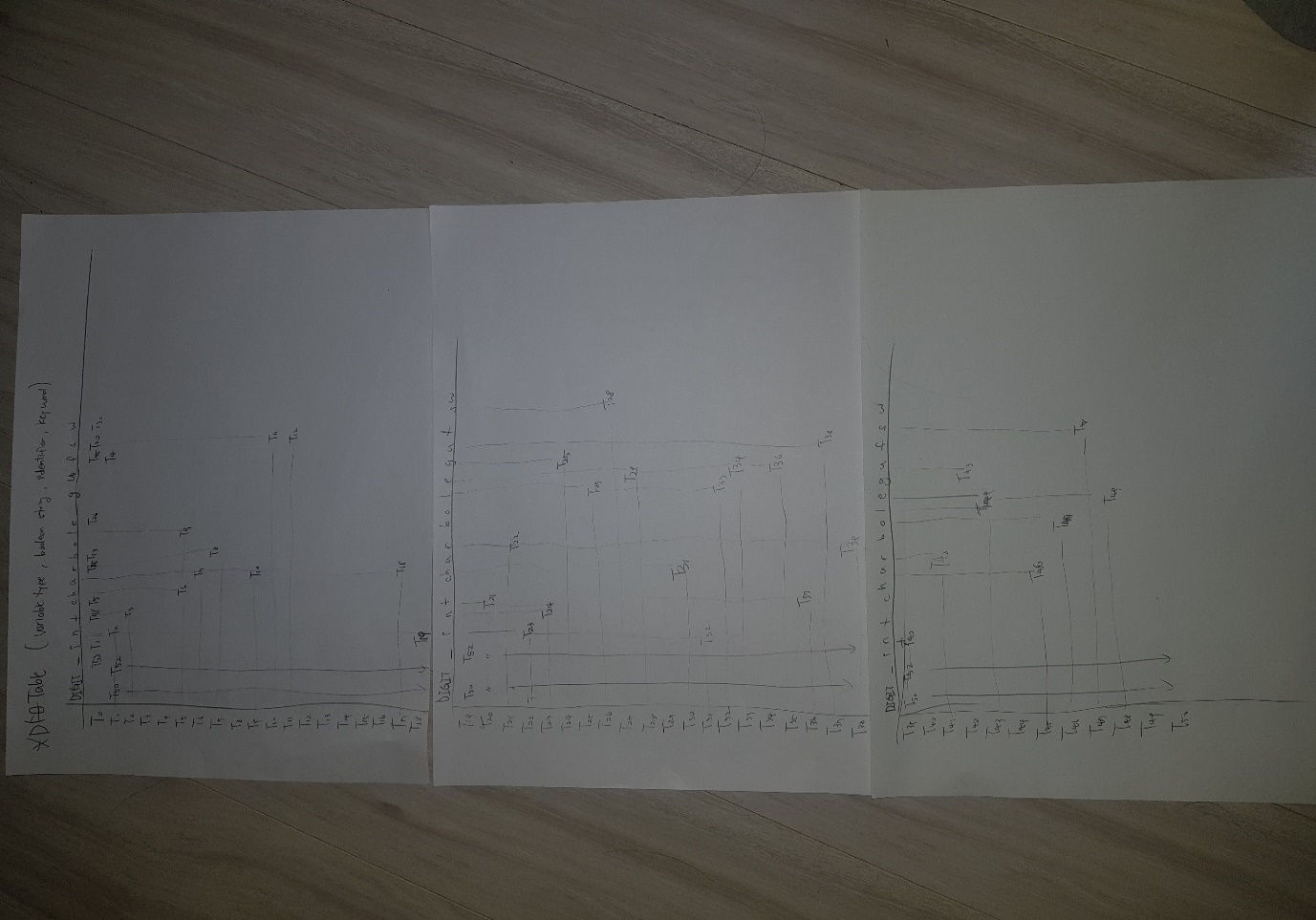
****

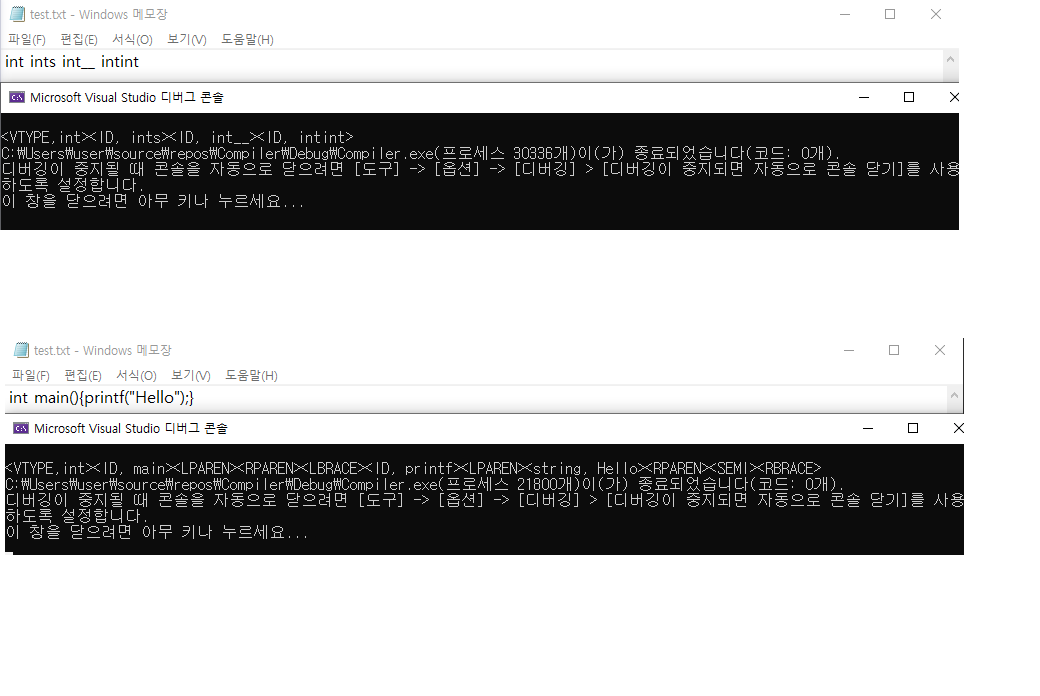
****

****

****

****

**3.**

**3.IMPLEMENTATION**

**4. DETAILS**

**- token을 구별하기 위한 방법**

**1. 토큰을 저장하는 배열과 파일출력을 받는 배열을 정의한다.**

**2. 파일을 getc로 읽어 하나하나의 문자로 배열에 저장한다.**

**3. 파일 내용이 저장된 배열을 앞에서부터 읽어 토큰이 만들어진다면 저장배열에 넣는다.**

**(토큰을 구분하는 방법은 코드에 설명되어 있다.)**

**4. 토큰을 넣은 후 바로 다음 칸에 공백문자를 집어넣는다.**

**이로써 배열에는 토큰’ ‘토큰’ ‘..이런 식으로 쌓여있다.**

**5. 따라서 나중에 토큰을 나눌 때는 공백문자를 기준으로 나눌 수 있다.**