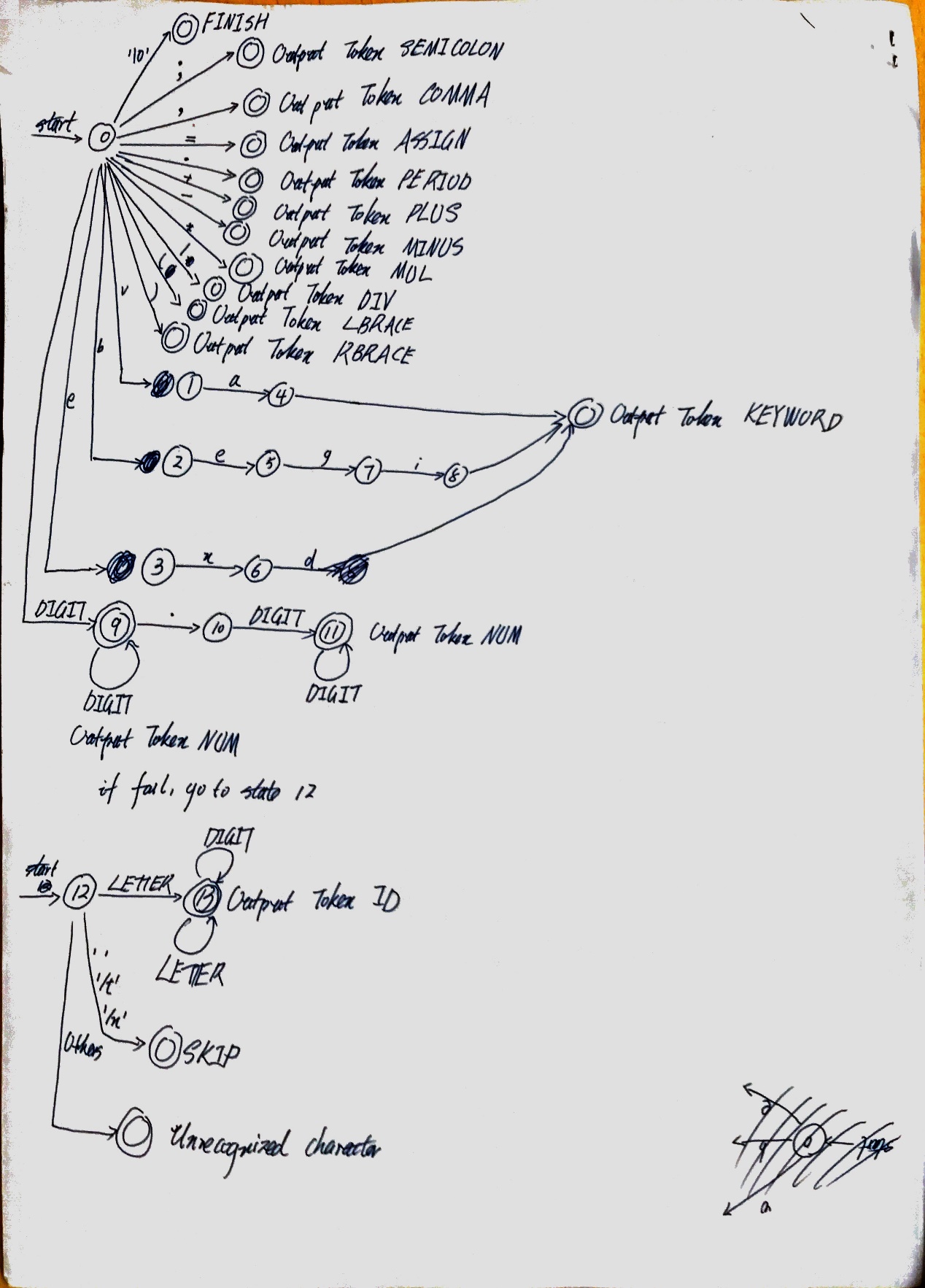
ZHENG Hongyi

13104036d

1. Regular Expressions
   1. VAR 🡪 var
   2. BEGIN 🡪 begin
   3. END 🡪 end
   4. KEYWORD 🡪 VAR | BEGIN | END
   5. COMMA 🡪 ,
   6. SEMICOLON 🡪 ;
   7. ASSIGN 🡪 =
   8. PERIOD 🡪 .
   9. DIGIT 🡪 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
   10. DIGITS 🡪 DIGIT (DIGIT)\*
   11. OPTIONAL\_FRAC 🡪 .DIGITS | ε
   12. NUM 🡪 DIGITS OPTIONAL\_FRAC
   13. PLUS 🡪 +
   14. MINUS 🡪 -
   15. MUL 🡪y \*
   16. DIV 🡪 /
   17. LBRACE 🡪 (
   18. RBRACE 🡪 )
   19. LETTER 🡪 A | B | …| Z | a | b | … | z
   20. ID 🡪 LETTER (LETTER | DIGIT ) \*
2. Finite Automata
3. Description of program
4. Description of functions

/\*----------Function: print\_token()--------------------

Print token based on definition and requirement

------------------------------------------------------\*/

/\*---------Function: fail(int cstate)----------------------------

Based on the current state, to decide the next state

current state < 12, next state <- 12 and start the second examination

else, next state = -1 and output unrecognized token

-------------------------------------------------------\*/

/\*--------Function: get\_next\_token()-----------------------

Pick each character iteratively

Transit between states according to the character value and the automata defined above

If an accepting state is found, return the next token.

Else, jump to state 12 and continue above processure

If fail also, output unrecognized character

Note: Return FINISH if normal finish;

Return ERROR if there is non-recognized a character

---------------------------------------------------------\*/

/\*-------------Function: scan() ---------------------------

Get each return token and call print\_token to print it

If FINISH is return, print out finish information

If ERROR is return, print out unrecognized input

----------------------------------------------------------\*/

1. How program works.
   1. Pick each character iteratively
   2. Start from state 0, and move forward according to each character
   3. If one of the accepting state is arrived, output the corresponding token
   4. Otherwise, jump to state 12
   5. Repeat b) and c)
   6. If fail, output ERROR
   7. If scanned all input, output FINISH