

04-F2

栈与队列

逆波兰表达式：转换

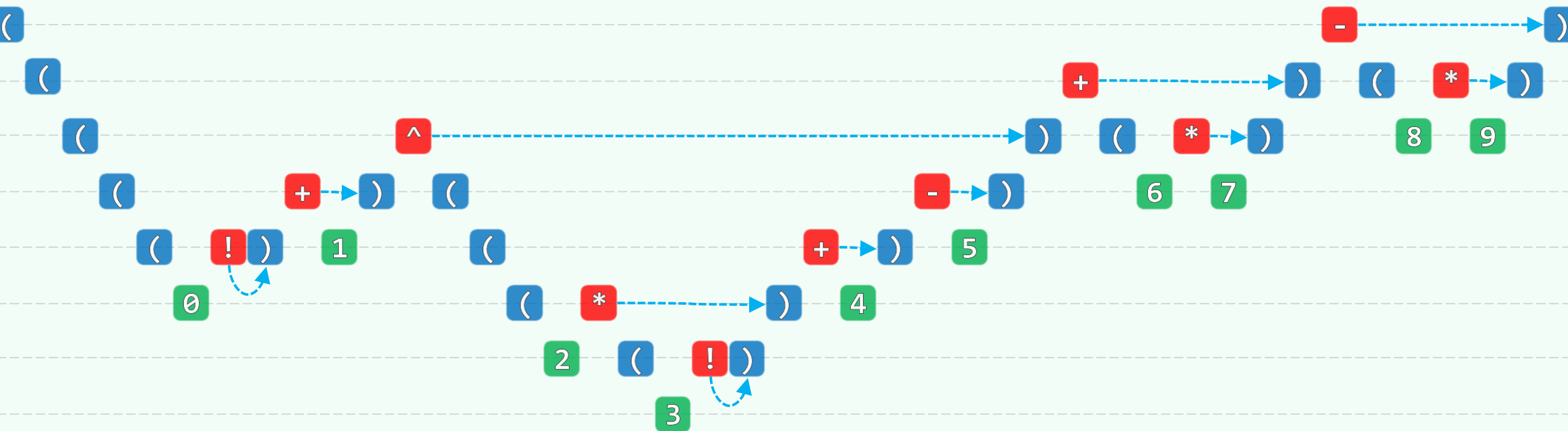
巨大的需求源于巨大的财富，而且要得到我们想要的，最好的办法往往是舍弃我们已拥有的

知與之為取，政之寶也

邓俊辉

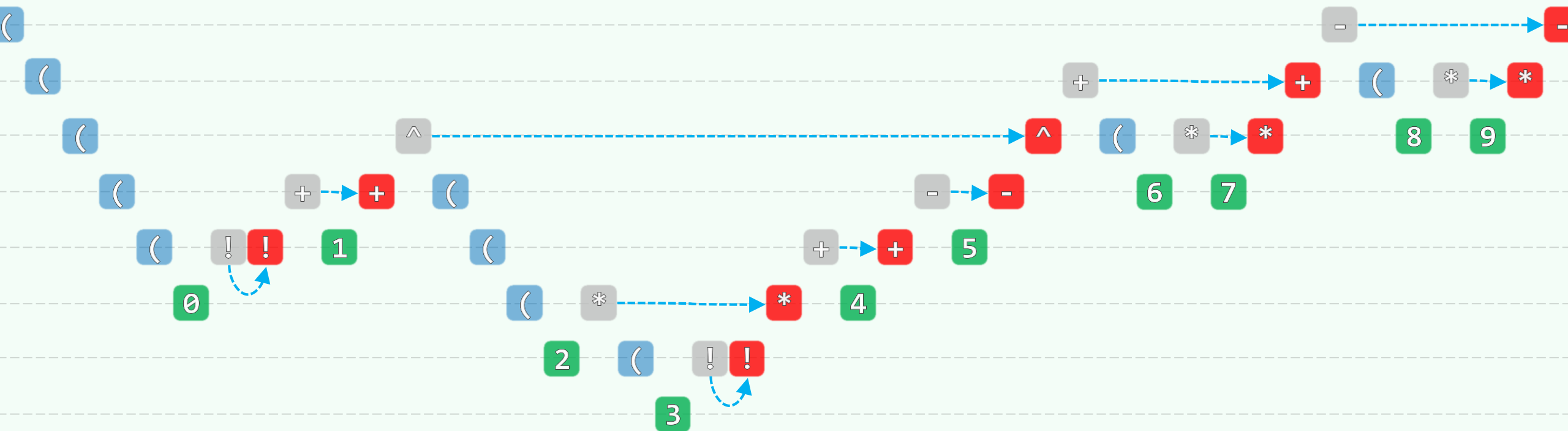
deng@tsinghua.edu.cn

手工转换：添加括号


$$(((((0!)+1)^((2*(3!))+4)-5))+((6*7))-((8*9)))$$

(0 ! + 1) ^ (2 * 3 ! + 4 - 5) + 6 * 7 - 8 * 9

手工转换：以运算符替换右括号，清除左括号



(((((0 ! ! + 1 + ^ (((2 * (3 ! ! * + 4 + - 5 - ^ + (6 * 7 * + - (8 * 9 * -
 0 ! 1 + 2 3 ! * 4 + 5 - ^ 6 7 * + 8 9 * -
 0 ! 1 + 2 3 ! * 4 + 5 - ^ 6 7 * + 8 9 * -

自动转换

```
double evaluate( char* S, char* RPN ) { //RPN转换
    /* ..... */
    while ( ! optr.empty() ) { //逐个处理各字符, 直至运算符栈空
        if ( isdigit( * S ) ) //若当前字符为操作数, 则直接
            { readNumber( S, opnd ); append( RPN, opnd.top() ); } //将其接入RPN
        else //若当前字符为运算符
            switch( priority( optr.top(), *S ) ) {
                /* ..... */
                case '>': { //且可立即执行, 则在执行相应计算的同时
                    char op = optr.pop(); append( RPN, op ); //将其接入RPN
                    /* ..... */
                } //case '>'
            }
        /* ..... */
    }
    /* ..... */
}
```