

CSCI 204: Data Structures & Algorithms

Stack Applications

TWO APPLICATIONS:

1. EVALUATION OF EXPRESSIONS
2. BACKTRACKING

Evaluating a postfix expression

```
# for simplicity, we may assume numbers are single digits represented as characters
# e.g., an expression may look like 2 3 + 3 4 * +
while expression not exhausted yet:
    read next token
    if it is a variable:
        push it onto the stack
    else: # an operator 'op'
        right = pop()
        left = pop()
        result = left op right # you should write a function for doing the arithmetic
        operation
        push the result back onto the stack
the value at the top of the stack is the result of the expression
```

Stacks

Backtracking – a stack application

Backtracking

- Here is an example: HPAir of using stack for backtracking in its scheduling of flights
 - Given
 - A set of cities that HPAir serves
 - Pairs of city names, each pair represents the origin and destination of one flight
 - Pair of cities names each of which represents a request to fly from an origin to a destination
 - Find whether or not a path exists between two cities requested by a passenger

General ideas

1. Start from the origin city
2. Find a city that is connected to the current city
3. If the next city is the destination, we are done. Report a route has been found
4. Otherwise go from this city and repeat step 2
5. If we visited all cities and no route is found, we declare the failure