

Recall: Word Equations

$$x \cdot y = u \cdot \text{'ab'} \cdot v$$

```
(assert (= (str.++ x y)
           (str.++ u "ab" v)))
```

Recall: Regular Languages

$$x \in L((ab)^* + c)$$

$$R ::= \varepsilon \mid a \mid R \cdot R \mid R + R \mid R^*$$

```
(assert (str.in_re x (re.++ (re.*  
    (str.to_re "ab"))) ...)))
```



<https://eldarica.org/ostrich-popl24/>

Creating a password

ostrich

Sorry, the password must be more than 7 characters.

running ostrich

Sorry, the password must contain at least one numerical character.

1 running ostrich

Sorry, the password cannot have blank spaces.

50BLOODYrunningostriches!

Sorry, the password must repeat itself twice.

50BLOODYrunningostriches!50BLOODYrunningostriches!

Sorry, that password is already in use!

Recall: Transducers

```
(define-fun-rec toUpper ((x String) (y String)) Bool
  (or (and (= x "") (= y ""))
      (and (not (= x "")) (not (= y ""))
            (= (char.code (str.head y))
                (ite (and (<= 97 (char.code (str.head x)))
                       (<= (char.code (str.head x)) 122))
                    (- (char.code (str.head x)) 32)
                    (char.code (str.head x))))
            (toUpper (str.tail x) (str.tail y))))
  )
```

Recall: Replace(-all)

- Replace first occurrences of u by v
 - $replace(x, u, v)$ (*str.replace* $x u v$)
- Replace all occurrences of u by v
 - $replaceAll(x, u, v)$ (*str.replace_all* $x u v$)

Generating Test Cases

```
def manipulate_string(input_string):  
    # Step 1: Replace 'p' with 'nuf'  
    replaced_string = input_string.replace('p', 'nuf')  
  
    # Step 2: Reverse the string  
    reversed_string = replaced_string[::-1]  
  
    # Step 3: Modify and concatenate additional text  
    additional_text = " This is fun! "  
    trimmed_additional_text = additional_text.strip() # Trim leading and trailing whitespace  
    modified_additional_text = trimmed_additional_text.upper() # Convert to uppercase  
  
    concatenated_string = reversed_string + modified_additional_text  
  
    # If condition: Check if the concatenated string contains "fun" (in lowercase)  
    if "fun" in concatenated_string:  
        print("The word 'fun' is in the final string.")
```

Exercises

1. Define the length constraint only using regular languages.
2. Use Ostrich to generate a password that satisfies all rules.
3. Add one rule such that the password must start with "popl" and generate a password using Ostrich.
4. Use word equations and string concatenation to delete the 3rd character from a string.
5. Define a transducer that converts characters in ASCII from upper case to lower case.