Python per la Manipolazione dei Testi

Parte 2

Cezar Sas
University of Groningen
c.a.sas@rug.nl

UNICODE

(e cenni sulle sue follie)

Categorie Unicode

- Caratteri Unicode (minuscolo/maiuscolo) - **Ll** **Lu**
- Operatori Matematici (= - + *)- **Sm**
- Caratteri Valuta ($ € £)- **Sc**
- ....

Usabili con:

```
pip install regex
import regex
```

https://www.compart.com/en/unicode/category

Categorie Unicode

- `[\p{L}--QW]` – Tutti i caratteri senza QW
- `[\p{N}--[0-9]]` – Tutti i caratteri numerici senza [0-9]
- `...`

https://www.compart.com/en/unicode/category

Ottimizzazione di Espressioni Regolari

(Tips & Tricks)

Backtracking

Comportamento di una regex quando fallisce. Ovvero arretra (backtracking) per provare a valutare possibili alternative.

Esempio:

r".+b" → acaaacbad

a**caaacbad** - .+

a**caaacbad** - .+

...

acaaacba**d** - .+

acaaacba**d** - .+

BACKTRACKING

acaaacba**d** - .+b

BACKTRACKING

acaaacb**ad** - .+b

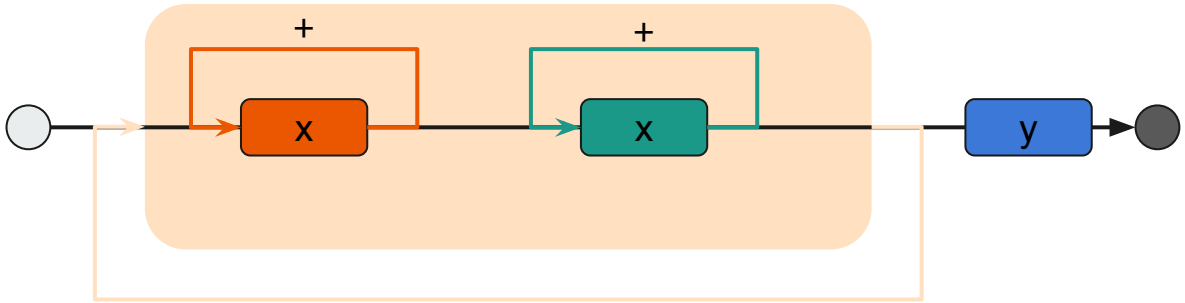
BACKTRACKING

acaaacb**ad** - .+b

Backtracking Catastrofico

r"(x+x+)+y"

xxxxxxxxxy

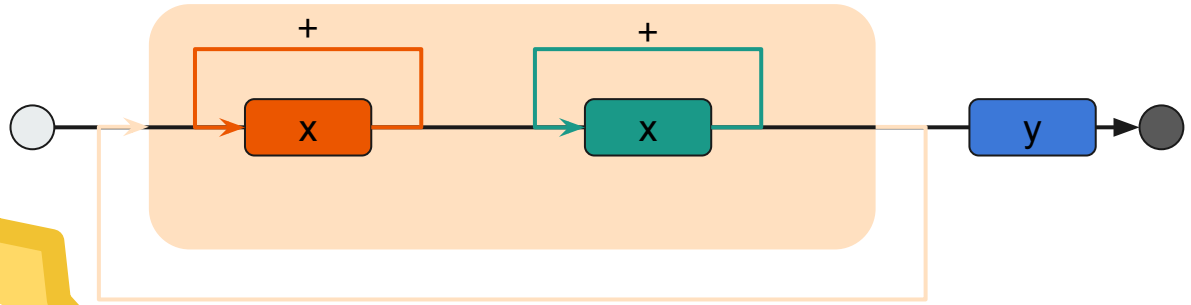


Adattato da: https://www.regular-expressions.info/catastrophic.html

Backtracking Catastrofico

r"(x+x+)+y"

xxxxxxxxxy



Adattato da: https://www.regular-expressions.info/catastrophic.html

Backtracking Catastrofico

Problema:

Quantificatori nested che hanno multiple opzioni di matching

Soluzione:

- Semplificare la regex
- *Quantificatori possessivi* e *gruppi atomici* (disponibili nel modulo `regex`)

Adattato da: https://www.regular-expressions.info/catastrophic.html

Tutto Opzionale

Esempio:

[-+]?[0-9]*\.?[0-9]*

Matcha anche il carattere vuoto (ε)

Soluzione:

[-+]?([0-9]*\.[0-9]+|[0-9]+)

Guida in dettaglio

https://www.regular-expressions.info/tutorial.html

LIVE! REC
CODING
(Soluzione Esercizi)