
WCC

Descrizione

WCC è una libreria **estensibile** che si occupa di **compilare** articoli wikipedia presenti nei dump **dumps.wikipedia.org**

Dependecy free

Wikipedia dump

```
{{short description|political movement that is
skeptical towards authority and rejects
involuntary, coercive hierarchy}}
{{redirect2|Anarchist|Anarchists|other
uses|Anarchists (disambiguation)}}
{{pp-move-indef}}
{{good article}}
{{use dmy dates|date=March 2020}}
{{use British English|date=January 2014}}
{{anarchism sidebar}}
{{basic forms of government}}
'''Anarchism''' is a [[political philosophy]] and
[[Political movement|movement]] that rejects all
involuntary, coercive forms of [[hierarchy]]. It
[[Radical politics|radically]] calls for the
abolition of
```

Perchè

1

Text extraction

La risoluzione del AST può essere utile per estrarre testo *pulito*

2

Text analysis

L'AST può essere analizzato per analizzare il testo e fare statistiche su di esso

3

Document preprocessing

Si presta bene a processi come quelli di l'indicizzazione di dump, in quanto è possibile selezionare contenuto specifico e indicizzare solo il contenuto che ha senso indicizzare

Markup spec

<!-- comment -->

{{template|param}}

[[link|display name]]

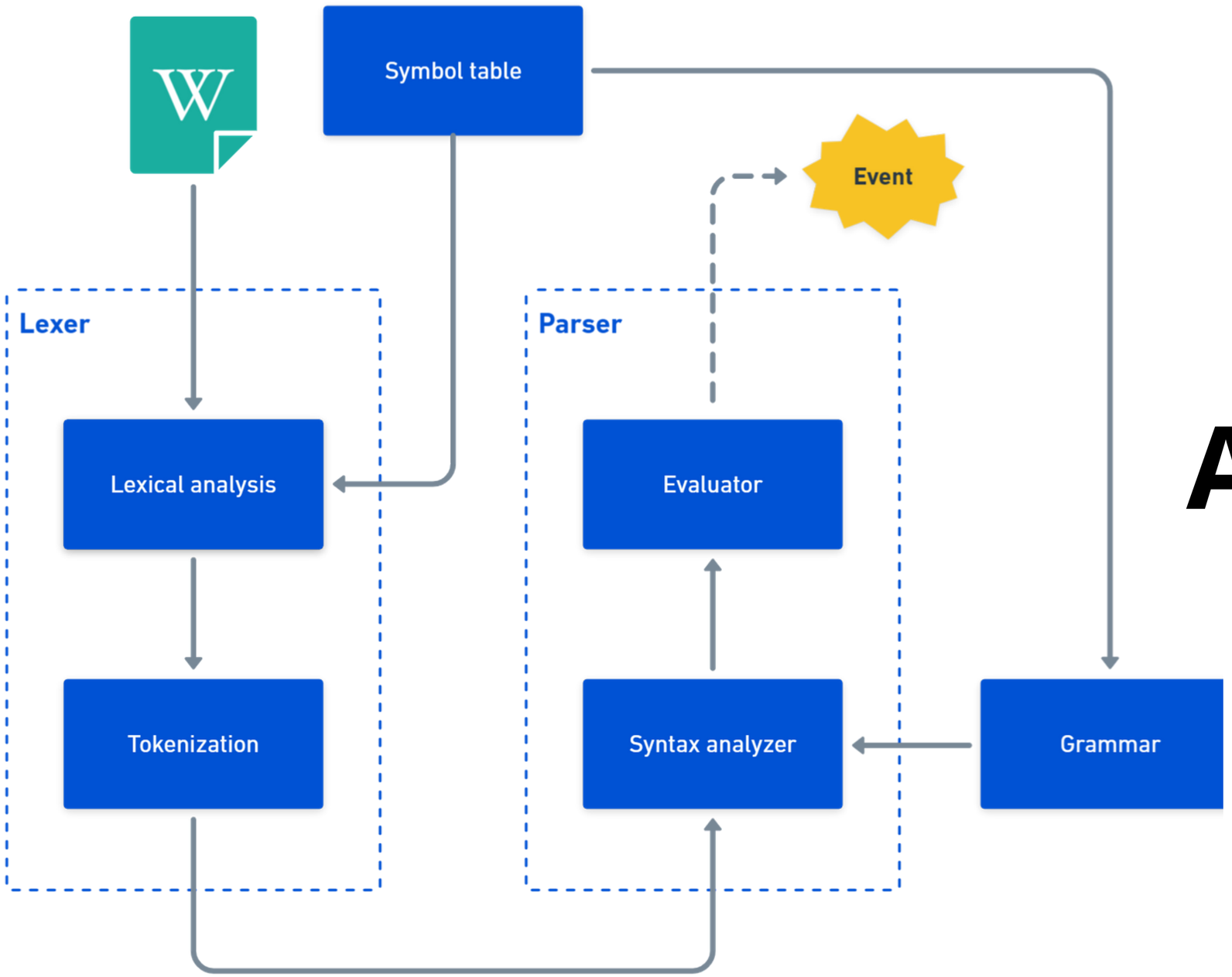
....e tanto altro

“

Wikitext, also known as Wiki markup or Wikicode, consists of the syntax and keywords used by the MediaWiki software to format a page.

en.wikipedia.org/wiki/Help:Wikitext

”



Architettura

Lexer

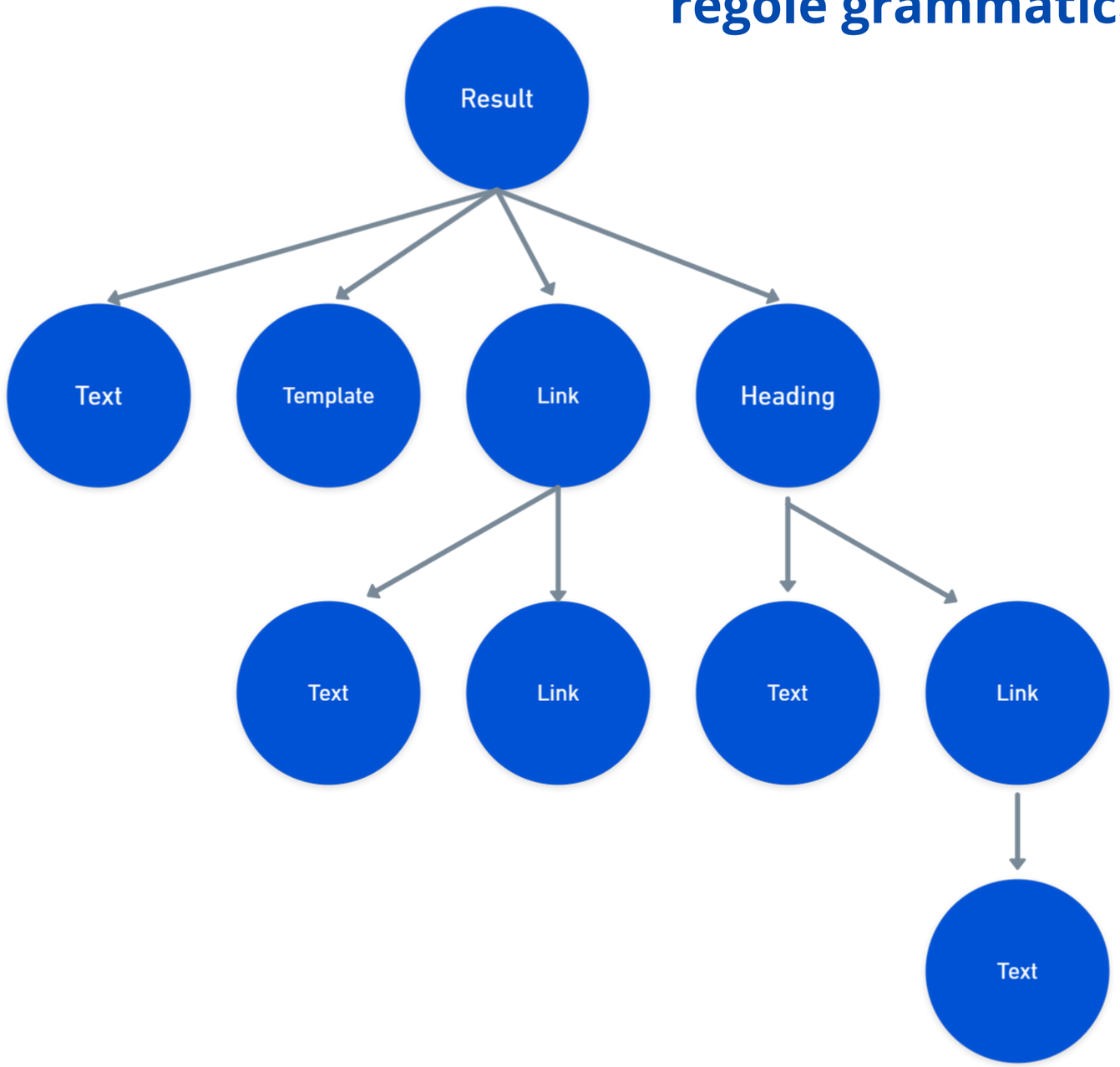
Il lexer è un automa a stati finiti che tramite espressioni regolari riconosce i simboli e da in output una sequenza di **token**. I simboli posso essere di tipo **RESERVED**, **IGNORE** o **ID**

Logs

```
[(TEMPLATE_START) [0], (TEXT) [2], (TEMPLATE_END) [63],
(LINE_BREAK) [64], (TEMPLATE_START) [65], (TEXT) [65],
(TEMPLATE_END) [65], (LINE_BREAK) [118], (TEMPLATE_START)
[119], (TEXT) [119], (TEMPLATE_END) [119], (LINE_BREAK)
[152], (TEMPLATE_START) [153], (TEXT) [153], (TEMPLATE_END)
[153], (LINE_BREAK) [1760], (LINE_BREAK) [2002], (TEXT)
[2003], (LINK_START) [2038], (TEXT) [2040], (LINK_END)
[2057], (TEXT) [2059], (LINK_START) [2112], (TEXT) [2114],
(LINK_END) [2127], (TEXT) [2129], (LINK_START) [2131],
(TEXT) [2133], (LINK_END) [2143], .... (EOF)]
```

Il parser esegue l'analisi sintattica e semantica seguendo precise **regole grammaticali.** Si tratta di un parser top-down **di tipo ricorsivo**

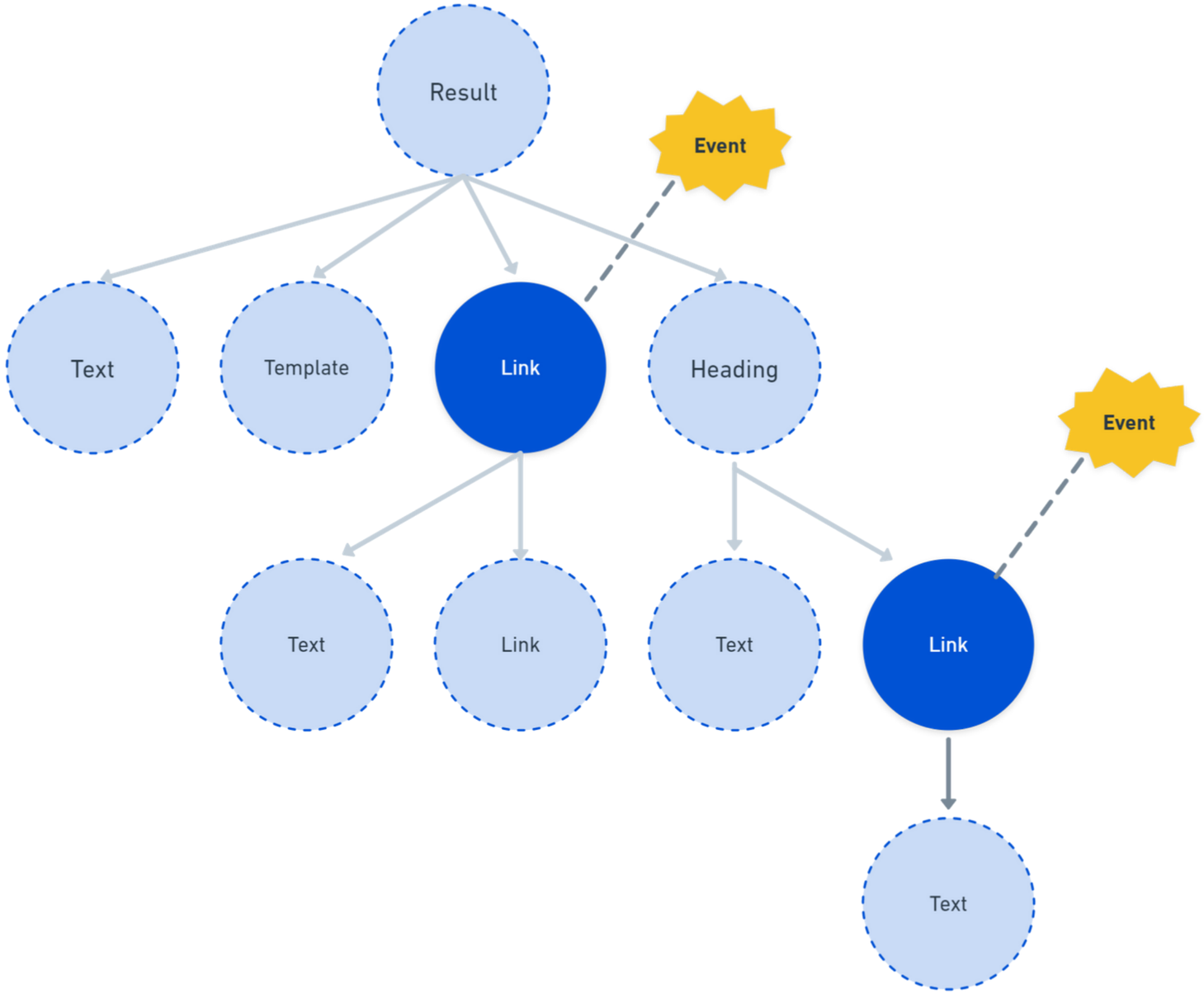
Parser



AST

Evaluation

È stata predisposta una fase di evaluation che si occupa di emettere eventi e renderizzare i nodi dell'AST



Evaluation

```
{{About|the band|their self-titled album|Dream
Theater (album)}}
{{short description|American progressive metal
band}}
{{Use mdy dates|date=April 2017}}
<!-- There's not much discussion going on the
talk page, so I've condensed the lead down by a
massive amount. Hopefully this will suffice. If
not, I've left the Lead Too Long tag here, just
commented out: {{Lead too long|date=June 2018}} -
->
'''Dream Theater''' is an American [[progressive
metal]] band formed in 1985 under the name
'''Majesty''' by [[John Petrucci]], [[John
Myung]] and [[Mike Portnoy]] while they attended
[[Berklee College of Music]] in [[Boston]],
[[Massachusetts]]. They subsequently dropped out
of their studies to concentrate further on the
band that would eventually become Dream Theater.
Though a number of lineup changes followed, the
three original members remained together until
September 8, 2010, when Portnoy left the band.
[[Mike Mangini]]
```

Dream Theater is an American progressive metal band formed in 1985 under the name Majesty by John Petrucci, John Myung and Mike Portnoy while they attended Berklee College of Music in Boston, Massachusetts. They subsequently dropped out of their studies to concentrate further on the band that would eventually become Dream Theater. Though a number of lineup changes followed, the three original members remained together until September 8, 2010, when Portnoy left the band. Mike Mangini

Grammar

> *So far progress has been made in both grammar definition and parser behaviour. Although none of the descriptions seems to be complete, some have achieved to describe a good part of the language.Current parser descriptions tried to do its best to follow the MediaWiki's parser behaviour.*

mediawiki.org/wiki/Markup_spec

Fonti differenti hanno grammatiche incomplete, ambigue o inconsistenti tra di loro

EBNF

```
start link    = "[[";
end link      = "]]";
internal link = start link, full
pagename, ["|", label], end link, label
extension;
```

ABNF

```
link = "[[" wikitext-L3 "]]"
wikitext-L3     = literal / template /
tplarg / link / comment /
                  line-eating-comment /
unclosed-comment / xmlish-element /
                  *wikitext-L3
```

Grammar

A seguito delle criticità evidenziate, si è scelto di definire una **grammatica interna per alcuni operatori (es Template)** che fosse la **meno ambigua, più completa e semplice possibile**

EBNF

```
template = "{{", title, { "|", part }, "}}" ;
part     = [ name, "=" ], value ;
title    = text ;
```

EBNF

```
text           :=  &
template       := '{{' text '}}'
```

Combinator Pattern

Si è usato un pattern di tipo combinatorio per scrivere **la grammatica del parser.**

Simple

Code reuse

Extensible

Python

```
seq(expect(Link.start),
    Grammar.epsilon,
    rep(sor(Grammar.epsilon, Grammar.template, Grammar.link, Grammar.linebreak),
    Link.end),
    expect(Link.end))
```

```
seq(expect(Template.start), Grammar.epsilon, expect(Template.end))
```

Python

```
def expect(token, consume=True):
    def parse(parser):
        current = parser.current
        if current and current.token == token:
            if consume:
                parser.next()
            return current
        return False

     return parse
```

Combinator Pattern

Statistiche

- **658MB**
- **19806 articoli**
- **4492 articoli redirect**
- **15204 articoli compilati con successo**
- **61 articoli con errori**

1.2GB

Peso indice senza compilazione

599MB

Peso indice con compilazione

Logs

Essendo una libreria i log vengono attivati **soltanto nella fase di testing.** Ogni modulo ha un proprio logger configurarabile da file

YAML

```yaml
version: 1
formatters:
  simple:
    format: '%(asctime)s - %(name)s[%(filename)s:%
(lineno)d] - %(levelname)s - %(message)s'
handlers:
  root:
    class: logging.FileHandler
    filename: 'default.log'
    formatter: simple
  lexer:
    class: logging.FileHandler
    filename: 'lexer.log'
    formatter: simple
  parser:
    class: logging.FileHandler
    filename: 'parser.log'
    formatter: simple
loggers:
  lexer:
    level: DEBUG
    handlers: [lexer]
    propagate: 0
  parser:
    level: DEBUG
    handlers: [parser]
    propagate: 0

root:
  level: INFO
  handlers: [root]
```


WCC