Parsing Contexts
Context Sensitive Grammars for Everyone

Jan Kurš
kurs@iam.unibe.ch
Software Composition Group
print "Hello!"

if (outOf(milk)):
    print "We are out of milk!"
    order(milk)

if (outOf(eggs)):
    ...
print “Hello!”

if (outOf(milk)):
    print “We are out of milk!”
    order(milk)

if (outOf(eggs)):
    ...
Print "Hello!"

if (outOf(milk)):
    print "We are out of milk!"
    order(milk)

if (outOf(eggs)):
    ...

ifKeyword := 'if' asParser.
print "Hello!"

if (outOf(milk)):
    print "We are out of milk!"
    order(milk)

if (outOf(eggs)):
    ...
print "Hello!"

if (outOf(milk)):
    print "We are out of milk!"
    order(milk)

if (outOf(eggs)):
    ...

identifier := #letter, (#letter / #digit) star.
Indentation using Petit Parser

print "Hello!"

if (outOf(milk)):
    print "We are out of milk!"
    order(milk)

if (outOf(eggs)):
Indentation using Petit Parser

```python
print "Hello!"
if (outOf(milk)):
    print "We are out of milk!"
    order(milk)
if (outOf(eggs)):
    ...

block := indent, command plus, dedent.
```
Indentation using Petit Parser

print “We are out of milk!”
order(milk)

block := indent, command plus, dedent.
print "We are out of milk!"
order(milk)

block := indent, command plus, dedent.
indent := ???
dedent := ???
State of the Art

- Regular Languages
  - Simple
- Context-Free Languages
- Context-Sensitive Languages

- Regular Expression

- Complex
State of the Art

- Turing Machine
- PetitParser
  Dynamic Grammars in Smalltalk
- Regular Expression
- Simple
- Complex
State of the Art

- Turing Machine
- PetitParser
  Dynamic Grammars in Smalltalk
- Regular Expression
  - Simple
  - Complex
State of the Art

- Turing Machine
- PetitParser
- Regular Expressions
- Simple
- Complex
State of the Art

- Context-Sensitive Languages
- Context-Free Languages
- Regular Languages

- Simple
- Complex

Turing Machine

PetitParser
Dynamic Grammars in Smalltalk

Regular Expression
Goal

PetitParser
Dynamic Grammars in Smalltalk
Parsing Contexts

Regular

Regular Expression

Simple

Complex

Turing Machine

Context-Sensitive

Context-Free

20
PetitParser uses **Parser Combinators**
PetitParser uses **Parser Combinators**

`#letter, (#letter / #digit) star`
PetitParser uses **Parser Combinators**

```plaintext
#letter, (#letter / #digit) star
```

PetitParser uses **Parser Combinators**

#letter, (#letter / #digit) star
String

String
String

String
String

"Hello"

#letter

"ello"
print "We are out of milk!"
order(milk)

block := indent, command plus, dedent.
print "We are out of milk!"
order(milk)

block := indent, command plus, dedent.
indent := ???
dedent := ???
Indentation Example

```
print "We are out of milk!"
order(milk)
```

```
block := indent, command plus, dedent.
```
Indentation Example

```
print "We are out of milk!"
order(milk)
```

```
block := indent, command plus, dedent.
indent := ???
dedent := ???
```
String

String
String

String

String & Indentation Stack

String & Indentation Stack
String

Parsing Contexts

String
print "We are out of milk!"
order(milk)
Regular Expression

Simple

Complex

Turing Machine

Regular

Context-Free

Context-Sensitive

PetitParser Dynamic Grammars in Smalltalk

Parsing Contexts
Grammar
Implementors
Grammar Implementors

Framework Implementors
Grammar Implementors

Framework Implementors

\[ N \leftarrow (N \cup T)^* \]
Grammar Implementors

Framework Implementors

\[ N \leftarrow (N \cup T)^* \]

indent, command plus, dedent
Grammar Implementors

\[ N \leftarrow (N \cup T)^* \]

indent, command plus, dedent

Framework Implementors

```plaintext
context | column indentation stack |
(\text{context:isBeginOfLine}) \text{ifFalse:} |
^ \text{Failure} |
\text{context consumeLeadingWhitespace}.

* Save the current column
  column := context stream column,
  stack := (context propertyAt: #indent).
  indentation := stack top.

(column > indentation) \text{ifTrue:} |
  stack push: column.
  ^ #indent
  ^ \text{Failure}
```

```plaintext
context | column referenceIndentation stack |
(\text{context:isBeginOfLine}) \text{ifFalse:} |
^ \text{Failure} |
\text{context consumeLeadingWhitespace}.

column := context stream column.

* Restore previous column from the *
  * stack and compare with current *
  stack := (context propertyAt: #indent).
  stack pop.
  referenceIndentation := stack top.

(column == referenceIndentation) \text{ifTrue:} |
  ^ #dedent
  ^ \text{Failure}
```
Regular

Regular Expression

Simple

Context-Free

Context-Sensitive

Complex

PetitParser
Dynamic Grammars in Smalltalk

Parsing Contexts

PetitParser
Dynamic Grammars in Smalltalk

Parsing Contexts

Turing Machine

46
indentation
indentation
tolerant parsing
indentation

tolerant parsing

debugging
String

Regular Expression

Turing Machine

Parsing Contexts

PetitParser

Dynamic Grammar in Smalltalk
\[ N \leftarrow (N \cup U T)^* \]

Grammar Implementor

Framework Implementors
```
N ← (N U T)*
```

**Grammar Implementor**

**Framework Implementors**

- indentation
- tolerant parsing
- debugging