Lekta framework practical tutorial

Jose F Quesada & Jose Luis Pro

Definition

Lekta is a **framework** oriented to the design and implementation of Natural Language Processing (NLP) related applications. This includes:

- Some specialized and **optimized** modules widely used in NLP applications (tokenizer, parser and so on). You'll never reinvent the wheel anymore.
- A simple and efficient way to define lexicons and grammar rules for any natural language.
- Early multilingual support for all your applications.
- A set of built-in functions that you'll find useful when implementing your NLP oriented app.
- A programming language to interact with all items above and to define your own functions or procedures.

Definition

Lekta is a **framework** oriented to the design and implementation of Natural Language Processing (NLP) related applications. This includes:

- Some specialized and **optimized** modules widely used in NLP applications (tokenizer, parser and so on). You'll never reinvent the wheel anymore.
- A simple and efficient way to define lexicons and grammar rules for any natural language.
- Early multilingual support for all your applications.
- A set of built-in functions that you'll find useful when implementing your NLP oriented app.
- A programming language to interact with all items above and to define your own functions or procedures.

Definition

Lekta is a **framework** oriented to the design and implementation of Natural Language Processing (NLP) related applications. This includes:

- Some specialized and **optimized** modules widely used in NLP applications (tokenizer, parser and so on). You'll never reinvent the wheel anymore.
- A simple and efficient way to define lexicons and grammar rules for any natural language.
- Early multilingual support for all your applications.
- A set of built-in functions that you'll find useful when implementing your NLP oriented app.
- A programming language to interact with all items above and to define your own functions or procedures.

Definition

Lekta is a **framework** oriented to the design and implementation of Natural Language Processing (NLP) related applications. This includes:

- Some specialized and **optimized** modules widely used in NLP applications (tokenizer, parser and so on). You'll never reinvent the wheel anymore.
- A simple and efficient way to define lexicons and grammar rules for any natural language.
- Early multilingual support for all your applications.
- A set of built-in functions that you'll find useful when implementing your NLP oriented app.
- A programming language to interact with all items above and to define your own functions or procedures.

Definition

Lekta is a **framework** oriented to the design and implementation of Natural Language Processing (NLP) related applications. This includes:

- Some specialized and **optimized** modules widely used in NLP applications (tokenizer, parser and so on). You'll never reinvent the wheel anymore.
- A simple and efficient way to define lexicons and grammar rules for any natural language.
- Early multilingual support for all your applications.
- A set of built-in functions that you'll find useful when implementing your NLP oriented app.
- A programming language to interact with all items above and to define your own functions or procedures.

Definition

Lekta is a **framework** oriented to the design and implementation of Natural Language Processing (NLP) related applications. This includes:

- Some specialized and **optimized** modules widely used in NLP applications (tokenizer, parser and so on). You'll never reinvent the wheel anymore.
- A simple and efficient way to define lexicons and grammar rules for any natural language.
- Early multilingual support for all your applications.
- A set of built-in functions that you'll find useful when implementing your NLP oriented app.
- A programming language to interact with all items above and to define your own functions or procedures.

One file .lkt and one file .slk

AnBm.lkt

```
1  // ********************************************************************
2  // Exercise 01: Generator/Recognizer for language AnBm. Where n,m >= 1
3  // ********************************************************************
4
5  lektaProject
6     projectHead
7        projectDescriptor : "AnBm"
8        projectLanguageScope : [ anbm ]
9        projectOperationScope : [ test ]
10       projectInterfaceScope : [ dialogue ]
11       projectCompileOutput : ".AnBm.olk"
12
13    projectSetup
14       setupParserRoots = S
15
16    classModel
17       classDef:Void ( S, A, B, a, b )
18
19    lexicalModel forLanguage anbm
20       ("a", a)
21       ("b", b)
22
23    grammaticalModel forLanguage anbm
24       (R1: [ S -> a A b B ])
25       (R2: [ A -> ])
26       (R3: [ A -> a A ])
27       (R4: [ B -> ])
28       (R5: [ B -> b B ])
```

Programming structures: comments

```
1 // This is a mono-line comment
```

```
1 /* This is a multi-line comment
2       with some lines
3       commented */
```

Programming structures: if...else if...else

```
1  if(cond1)
2  {
3      // Body 1
4  }
5  else if(cond2)
6  {
7      // Body 2
8  }
9  ...
10 else
11 {
12     // Body n
13 }
```

Programming structures: switch

```
1 // TODO
```

Programming structures: cond

```
1 // TODO
```

Programming structures: while

```
1 // TODO
```

Programming structures: for

```
1 // TODO
```

Opperators:

```
1 // TODO
```