# ElixirFM Functional Arabic Morphology

Otakar Smrž

Institute of Formal and Applied Linguistics
Charles University in Prague

إِجتِمَاعُ خُبَرَاءِ المُحَلِّلَاتِ الصَّرفِيَّةِ الحَاسُوبِيَّةِ لِلُّغَةِ العَرَبِيَّةِ

مَجمَعُ اللُّغَةِ العَرَبِيَّةِ بِدِمَشقَ

April 27, 2009

## Introduction

ElixirFM is an implementation of a novel computational model of the morphological processes in Modern Written Arabic. ElixirFM is related to the Prague Arabic Dependency Treebank project.

## Introduction

ElixirFM is an implementation of a novel computational model of the morphological processes in Modern Written Arabic. ElixirFM is related to the Prague Arabic Dependency Treebank project.

The core of ElixirFM is written in Haskell, while interfaces in Perl support lexicon editing and other interactions. The essential components of the system include a multi-purpose programming library promoting clear style and abstraction in the model, and a linguistically refined, yet intuitive and efficient, morphological lexicon.

## Introduction

ElixirFM is an implementation of a novel computational model of the morphological processes in Modern Written Arabic. ElixirFM is related to the Prague Arabic Dependency Treebank project.

The core of ElixirFM is written in Haskell, while interfaces in Perl support lexicon editing and other interactions. The essential components of the system include a multi-purpose programming library promoting clear style and abstraction in the model, and a linguistically refined, yet intuitive and efficient, morphological lexicon.

Definition of lexemes includes the derivational root and pattern information if appropriate. Modeling of the written language as well as spoken dialects is expected methodologically identical.

## Characteristics

ElixirFM is inspired by the Functional Morphology library for Haskell.

The lexicon of ElixirFM is derived from the open-source Buckwalter lexicon, but it is redesigned in important respects and extended.

## Characteristics

ElixirFM is inspired by the Functional Morphology library for Haskell.

The lexicon of ElixirFM is derived from the open-source Buckwalter lexicon, but it is redesigned in important respects and extended.

Morphology is modeled in terms of abstract patterns, paradigms, grammatical categories, lexemes, and word classes. The computation involved in analysis or generation is conceptually distinguished from the general-purpose linguistic model.

## Characteristics

ElixirFM is inspired by the Functional Morphology library for Haskell.

The lexicon of ElixirFM is derived from the open-source Buckwalter lexicon, but it is redesigned in important respects and extended.

Morphology is modeled in terms of abstract patterns, paradigms, grammatical categories, lexemes, and word classes. The computation involved in analysis or generation is conceptually distinguished from the general-purpose linguistic model.

Word forms are represented in an abstract and extensible notation encoding both orthography and phonology, like in ArabTEX.

## Characteristics

ElixirFM is inspired by the Functional Morphology library for Haskell.

The lexicon of ElixirFM is derived from the open-source Buckwalter lexicon, but it is redesigned in important respects and extended.

Morphology is modeled in terms of abstract patterns, paradigms, grammatical categories, lexemes, and word classes. The computation involved in analysis or generation is conceptually distinguished from the general-purpose linguistic model.

Word forms are represented in an abstract and extensible notation encoding both orthography and phonology, like in ArabTeX.

`"al-lA-silkIyu"` *al-lā-silkīyu* اللاسلكي اَللَّاسِلكِيُّ اَللَّاسِلْكِيُّ

## Characteristics

ElixirFM is inspired by the Functional Morphology library for Haskell.

The lexicon of ElixirFM is derived from the open-source Buckwalter lexicon, but it is redesigned in important respects and extended.

Morphology is modeled in terms of abstract patterns, paradigms, grammatical categories, lexemes, and word classes. The computation involved in analysis or generation is conceptually distinguished from the general-purpose linguistic model.

Word forms are represented in an abstract and extensible notation encoding both orthography and phonology, like in ArabTEX.

`"al-lA-silkIyu"`   *al-lā-silkīyu*   اللاسلكي   اَللَّاسِلكِيُّ   اَللَّاسِلْكِيُّ

```
"s l k"    `merge`    al >| lA >| FiCL |< Iy |<< "u"
```

```
|> "s l k" <| [
    FaCaL                    `verb`    [ "proceed", "behave" ]
        `imperf`  FCuL,
    FiCL                     `noun`    [ "wire", "thread" ]
        `plural`  HaFCAL,
    FiCL |< Iy               `adj`     [ "wire", "by wire" ],
    lA >| FiCL |< Iy         `adj`     [ "wireless", "radio" ],
    FuCUL                    `noun`    [ "behavior", "conduct" ],
    FuCUL |< Iy              `adj`     [ "behavioral" ],
    MaFCaL                   `noun`    [ "road", "method" ]
        `plural`  MaFACiL                                      ]
```

```
|> "s l k" <| [
    FaCaL                 `verb`    [ "proceed", "behave" ]
        `imperf`  FCuL,
    FiCL                  `noun`    [ "wire", "thread" ]
        `plural`  HaFCAL,
    FiCL |< Iy            `adj`     [ "wire", "by wire" ],
    lA >| FiCL |< Iy      `adj`     [ "wireless", "radio" ],
    FuCUL                 `noun`    [ "behavior", "conduct" ],
    FuCUL |< Iy           `adj`     [ "behavioral" ],
    MaFCaL                `noun`    [ "road", "method" ]
        `plural`  MaFACiL                                      ]
```

| | | | |
|---|---|---|---|
| proceed, behave | I(*u*) *salak* سَلَك | behavior, conduct | *sulūk* سُلُوك |
| wire, thread | (*ʾaslāk* أَسلَاك) *silk* سِلك | behavioral | *sulūkīy* سُلُوكِيّ |
| wire, by wire | *silkīy* سِلكِيّ | road, method | *maslak* مَسلَك |
| wireless, radio | *lā-silkīy* لَاسِلكِيّ | | (*masāliku* مَسَالِكُ) |

| | | |
|---|---|---|
| Form | All~Asilokiy~apu | اللّاسِلْكِيَّةُ |
| Morph | Al + lAsilokiy~ + ap + u | |
| Tag | DET+ADJ+NSUFF_FEM_SG+CASE_DEF_NOM | |
| Gloss | the + wireless / radio + (fem.sg.) + (def.nom.) | |
| Lemma | lAsilokiy~_1 | لَاسِلْكِيّ |

| | | |
|---|---|---|
| Form | All~Asilokiy~apu | اللّاسِلْكِيَّةُ |
| Morph | Al + lAsilokiy~ + ap + u | |
| Tag | DET+ADJ+NSUFF_FEM_SG+CASE_DEF_NOM | |
| Gloss | the + wireless / radio + (fem.sg.) + (def.nom.) | |
| Lemma | lAsilokiy~_1 | لاسِلْكِيّ |

.................................................................................................

| | | |
|---|---|---|
| Form | *al-lA-silkIyaTu* | *al-lā-silkīyatu* اَللّاسِلْكِيَّةُ |
| Morph | al >\| lA >\| FiCL \|< Iy \|< aT \|<< *"u"* | |
| Tag | A-----FS1D | |
| Form | *lA-silkIy* | *lā-silkīy* لَاسِلْكِيّ |
| Morph | lA >\| FiCL \|< Iy | |
| Root | *"s l k"* | |
| Reflex | wireless, radio | |
| Class | adjective | |

|  |  |  |
|---|---|---|
| Form | waOuxoraY | وَأُخْرَى |
| Morph | wa + OuxoraY |  |
| Tag | CONJ+ADJ |  |
| Gloss | and + other / another / additional |  |
| Lemma | OuxoraY_1 | أُخْرَى |

| | | |
|---|---|---|
| Form | waOuxoraY | وَأُخْرَى |
| Morph | wa + OuxoraY | |
| Tag | CONJ+ADJ | |
| Gloss | and + other / another / additional | |
| Lemma | OuxoraY_1 | أُخْرَى |

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

| | | | | | |
|---|---|---|---|---|---|
| Form | 'u_hrY | ʾuḫrā أُخْرَى | wa | wa وَ |
| Morph | FuCLY \|<< "u" | | "wa" | |
| Tag | A-----FS1I | | C---------- | |
| Form | 'A_har | ʾāḫar آخَر | wa | wa وَ |
| Morph | HACaL | | "wa" | |
| Root | "' _h r" | | "w" | |
| Reflex | other, another | | and | |
| Class | adjective | | conjunction | |

| | | |
|---|---|---|
| Form | `sayad~aEiy` | سَيَدَّعِي |
| Morph | `sa` + `ya` + `d~aEiy` + `(null)` | |
| Tag | `FUT+IV3MS+IV+IVSUFF_MOOD:I` | |
| Gloss | will + he / it + allege / claim / testify + (ind.) | |
| Lemma | `Aid~aEaY_1` | اِدَّعَى |

| | | |
|---|---|---|
| Form | sayad~aEiy | سَيَدَّعِي |
| Morph | sa + ya + d~aEiy + (null) | |
| Tag | FUT+IV3MS+IV+IVSUFF_MOOD:I | |
| Gloss | will + he / it + allege / claim / testify + (ind.) | |
| Lemma | Aid~aEaY_1 | اِدَّعَى |

| | | | | |
|---|---|---|---|---|
| Form | yadda'I | *yaddaʿī* يَدَّعِي | sa | *sa* سَ |
| Morph | "ya" >>\| FtaCI \|<< "u" | | "sa" | |
| Tag | VIIA-3MS-- | | F--------- | |
| Form | idda'Y | *iddaʿā* اِدَّعَى | sa | *sa* سَ |
| Morph | IFtaCY | | "sa" | |
| Root | "d ' w" | | "s" | |
| Reflex | allege, claim, testify | | *future marker* | |
| Class | verb | | particle | |

ElixirFM carefully designs the morphophonemic patterns of the templates, along with the phonological rules hidden in the `>|` or `|<<` operators. This greatly simplifies the morphological rules proper, inflectional or derivational. ElixirFM implements many generalizations of classical grammars, and suggests some new ones.

| | | |
|---|---|---|
| `"ya" >>| FtaCI |<< "u"` | `yadda'I` | yaddaʿī يَدَّعِي |
| `"ya" >>| FtaCI |<< "a"` | `yadda'iya` | yaddaʿiya يَدَّعِيَ |
| `"ya" >>| FtaCI |<< ""` | `yadda'i` | yaddaʿi يَدَّعِ |

ElixirFM carefully designs the morphophonemic patterns of the templates, along with the phonological rules hidden in the `>|` or `|<<` operators. This greatly simplifies the morphological rules proper, inflectional or derivational. ElixirFM implements many generalizations of classical grammars, and suggests some new ones.

| | | |
|---|---|---|
| `"ya" >>\| FtaCI \|<< "u"` | `yadda'I` | *yaddaʿī* يَدَّعِي |
| `"ya" >>\| FtaCI \|<< "a"` | `yadda'iya` | *yaddaʿiya* يَدَّعِيَ |
| `"ya" >>\| FtaCI \|<< ""` | `yadda'i` | *yaddaʿi* يَدَّعِ |
| `"ya" >>\| FCuL \|<< "u"` | `yaktubu` | *yaktubu* يَكْتُبُ |
| `"ya" >>\| FCuL \|<< "a"` | `yaktuba` | *yaktuba* يَكْتُبَ |
| `"ya" >>\| FCuL \|<< ""` | `yaktub` | *yaktub* يَكْتُبْ |

## Functionality

ElixirFM implements various user-end functions for lookup in the lexicon

## Functionality

ElixirFM implements various user-end functions for lookup in the lexicon, inflection and derivation of lexemes

## Functionality

ElixirFM implements various user-end functions for lookup in the lexicon, inflection and derivation of lexemes, resolution of strings

## Functionality

ElixirFM implements various user-end functions for lookup in the lexicon, inflection and derivation of lexemes, resolution of strings, exporting and pretty-printing of the information, et cetera.

## Functionality

ElixirFM implements various user-end functions for lookup in the lexicon, inflection and derivation of lexemes, resolution of strings, exporting and pretty-printing of the information, et cetera.

```
lookup (lA >| FiCL |< Iy)    lookup "lA-silkIy"    lookup "لاسلكي"

 lookup (words "wireless")      lookup (words "village school")
```

## Functionality

ElixirFM implements various user-end functions for lookup in the lexicon, inflection and derivation of lexemes, resolution of strings, exporting and pretty-printing of the information, et cetera.

```
lookup (lA >| FiCL |< Iy)   lookup "lA-silkIy"   lookup "لاسلكي"
 lookup (words "wireless")     lookup (words "village school")

     inflect (lA >| FiCL |< Iy `adj` []) "------F[SP]-D"
```

## Functionality

ElixirFM implements various user-end functions for lookup in the lexicon, inflection and derivation of lexemes, resolution of strings, exporting and pretty-printing of the information, et cetera.

```
lookup (lA >| FiCL |< Iy)   lookup "lA-silkIy"   lookup "لاسلكي"
 lookup (words "wireless")     lookup (words "village school")

     inflect (lA >| FiCL |< Iy `adj` []) "------F[SP]-D"

 derive ("w .s y" <-> HaFCY `verb` ["recommend"]) "A--P------"
```

## Functionality

ElixirFM implements various user-end functions for lookup in the lexicon, inflection and derivation of lexemes, resolution of strings, exporting and pretty-printing of the information, et cetera.

```
lookup (lA >| FiCL |< Iy)   lookup "lA-silkIy"   lookup "لاسلكي"

 lookup (words "wireless")      lookup (words "village school")

      inflect (lA >| FiCL |< Iy `adj` []) "------F[SP]-D"

  derive ("w .s y" <-> HaFCY `verb` ["recommend"]) "A--P------"

        resolve "mU.saNY bihi"     resolve "mūṣan bihi"
          resolve "مُوصًى بِهِ"     resolve "موصى به"
```

## Interfaces

There are various interfaces to ElixirFM, ranging from command-line interpreters or executables up to graphical linguistic annotation environments or user-friendly web applications, like the recently published ElixirFM Online Interface.

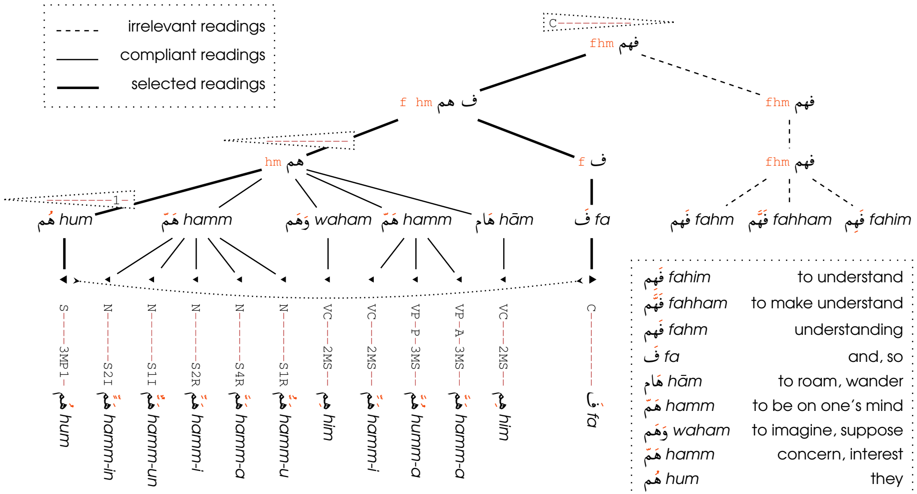
## Interfaces

There are various interfaces to ElixirFM, ranging from command-line interpreters or executables up to graphical linguistic annotation environments or user-friendly web applications, like the recently published ElixirFM Online Interface.

The TrEd tree editor is designed and implemented by Petr Pajas, with numerous annotation contexts contributed by other authors. Examples of our work include the ElixirFM and MorphoTrees contexts, or miscellaneous conversion templates or scripts for error detection and consistency checking.

`http://ufal.mff.cuni.cz/~pajas/tred/`

# Multi-Modal Annotation



| | |
|---|---|
| فَهِم *fahim* | to understand |
| فَهَّم *fahham* | to make understand |
| فَهْم *fahm* | understanding |
| فَ *fa* | and, so |
| هَام *hām* | to roam, wander |
| هَمّ *hamm* | to be on one's mind |
| وَهَم *waham* | to imagine, suppose |
| هَمّ *hamm* | concern, interest |
| هُم *hum* | they |

## Prague Arabic Dependency Treebank

PADT is a project of linguistic annotation of Modern Written Arabic based on the theory of Functional Generative Description.

## Prague Arabic Dependency Treebank

PADT is a project of linguistic annotation of Modern Written Arabic based on the theory of Functional Generative Description.

PADT consists mainly of the morphological and analytical levels of description. The annotation of tectogrammatics and information structure is initiated, as are valency frames in the new lexicon.

## Prague Arabic Dependency Treebank

PADT is a project of linguistic annotation of Modern Written Arabic based on the theory of Functional Generative Description.

PADT consists mainly of the morphological and analytical levels of description. The annotation of tectogrammatics and information structure is initiated, as are valency frames in the new lexicon.

PADT 1.0 was published in 2004 and has been used by tens of academic and commercial institutions.

## Prague Arabic Dependency Treebank

PADT is a project of linguistic annotation of Modern Written Arabic based on the theory of Functional Generative Description.

PADT consists mainly of the morphological and analytical levels of description. The annotation of tectogrammatics and information structure is initiated, as are valency frames in the new lexicon.

PADT 1.0 was published in 2004 and has been used by tens of academic and commercial institutions.

PADT 2.0 is due in 2009 and will cover over one million words of text. It merges original Prague Arabic Dependency Treebank annotations with converted and enhanced Penn Arabic Treebank.

## References

ElixirFM plus lexicons, Encode Arabic, MorphoTrees, and ArabTEX extensions are open-source software licensed under GNU GPL:

`http://sourceforge.net/projects/elixir-fm/`

`http://sourceforge.net/projects/encode-arabic/`

## References

ElixirFM plus lexicons, Encode Arabic, MorphoTrees, and ArabTeX extensions are open-source software licensed under GNU GPL:

`http://sourceforge.net/projects/elixir-fm/`

`http://sourceforge.net/projects/encode-arabic/`

ElixirFM Online Interface is the multi-modal user-end application:

`http://quest.ms.mff.cuni.cz/elixir/`

`http://ufal.mff.cuni.cz/padt/online/`

- Buckwalter, Tim. Buckwalter Arabic Morphological Analyzer 1.0. LDC2002L49, ISBN 1-58563-257-0. 2002
- Forsberg, Markus and Aarne Ranta. Functional Morphology. Proceedings of ICFP 2004, pages 213–223. ACM Press. 2004
- Lagally, Klaus. ArabTEX: Typesetting Arabic and Hebrew, User Manual Version 4.00. Technical Report 2004/03, Fakultät Informatik, Universität Stuttgart. 2004
- Smrž, Otakar. Functional Arabic Morphology. Formal System and Implementation. Ph.D. thesis, Charles University in Prague. 2007
- Smrž, Otakar et al. Prague Arabic Dependency Treebank: A Word on the Million Words. LREC 2008 Workshop on Arabic and Local Languages. 2008