July 1, 2004

ISO TC 37/SC 4 SC4N119.pdf Rev. 2

Language Resource Management

Descriptors and Mechanisms for Language Resources

Title:	Draft - Language Resource Management – Morpho-Syntactic Annotation Framework (MAF)
Editor(s):	Lionel Clément & Eric de la Clergerie
Source:	WG2
Project number:	NP 24611
Status:	WD
Date:	July 1, 2004
Agenda/Action:	For Review
References:	

Mr. Key-Sun Choi - SC4 Secretary - KORTERM - 373-1 Guseong-dong Yuseong-gu -
Daejeon 305-701 - Korea

Tel: +82 42 869 35 25 - Fax: +82 42 869 87 90 - kschoi@cs.kaist.ac.kr -
http://tc37sc4.org

Language Resource Management – Morpho-Syntactic Annotation Framework (MAF)

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Language Resource Management –
Morpho-Syntactic Annotation Framework (MAF)

Introduction

TO BE COMPLETED

Scope

In Natural Language Processing (NLP), the morpho-syntactic annotation phase assigns to each document segment (either text or speech) one or more *tags* providing information about the *part of speech* (noun, adjective, verb, . . .), morphological and grammatical features (such as number, gender, person, mode, verbal tense, . . .) and possibly other specific linguistic properties. The morpho-syntactic annotations attached to a segment do not refer to other segments or annotations, even if the choice of an annotation may depend on the surrounding context.

Normative references

- Data Category Registry (DCR)
- Feature Structure Representation (FSR) and Feature Structure Declaration (FSD)
- Linguistic Annotation Framework (LAF)
- Text Encoding Initiative (TEI) – Chapters to be precised
- MPEG7 – About referring positions in multimedia documents

Terms and definitions

Associative relation

Relations by which a linguistic unit is associated with others. This is a mental association which does not requires their effective presence.[1]

DAG - Directed Acyclic Graph TO BE COMPLETED

FSA - Finite State Automata TO BE COMPLETED See DAG.

Lemma

Class of inflected forms differing only by inflectional morphology. A lemma is usually refered to by one of these forms, arbitrarily chosen (e.g. infinitive for french verbs).

Lexeme

Lexical **morpheme**. Distinguished from a grammatical morpheme by the fact that it belongs to an open list and that it bears an autonomous signification.

[1] It differs from a paradigmatic relation because the latter only refers to linguistic units associated by substitutability.

Morpheme

Smallest linguistic entity bearing a signification in a discourse. A morpheme is either grammatical (grammeme) or lexical (lexeme).

Natural Language Processing

Field covering knowledge and techniques allowing computerized processing of linguistic data.

This field requests skills related, among others, to linguistics, mathematical logic, statistics, and algorithmics.

Syntagmatic relation

Relations by which linguistic units present in the discourse are associated.

Morpho-syntactic tag

To an associative relation corresponds a feature, for which the related entities share the same value. The morpho-syntactic tag lists some of these features (part-of-speech, grammatical category, etc.).

Token

Connex and not-empty discourse sequence identified as such by a morpho-phonologic analysis or an automatic processing of the discourse.

This can involve the recognition of a regular or algebric language (matching of the separators), or a lexicological analysis (recognition of roots, morphological derivation and inflection, etc.).

Morpho-syntactic unit - Word-form

Connex or not-connex entity from a speech sequence identified as such in an **associative relation**. This identification is the basis of morpho-syntactic tagging (part-of-speech, grammatical category, agreement feature, etc.). Morpho-syntactic units may have no acoustic or graphic realization, or correspond to one or more **tokens**.

Word Lattice TO BE COMPLETED See DAG.

Contents

1 General characteristics

1.1 Overview

In the Linguistic community, Morpho-Syntactic Annotations provide an important layer of linguistic information to a document, even if they do not cover the full range of possible linguistic annotations. For instance, other kind of annotations on references, discourse, prosody, or parsing may complete morpho-syntactic annotations.

Syntax and semantic can not be avoided in the definition of the parts of speech and of the grammatical categories. For instance, pronouns and substantives intrinsically carry a reference (to some entity); the tense or the aspect of verbs indicate the temporal deixis; the person, modality and other grammatical categories indicate the enunciation situation, ...

Therefore, it is not easy to provide an exact and precise definition of what cover morpho-syntactic annotations because they are strongly related to many other linguistic properties for a given language and for a given context.

Nevertheless, the present proposal tries to delimit minimal and maximal sequences in documents (either text or speech) that can be identified as *morpho-syntactic units* and tries to categorize the linguistic properties that may be used to mark these units, within some larger syntagmatic context. Minimal units can not be broken into subparts that could be identified by similar morpho-syntactic criteria, but may however still be broken into more atomic subparts with morphological or phonological properties. Morpho-syntactic units can be nested to form maximal units (such as compound words) that act as elementary units for other level of linguistic analysis, in particular parsing. The exact boundary between morpho-syntax and parsing may sometimes be fuzzy.

1.2 Meta-Model

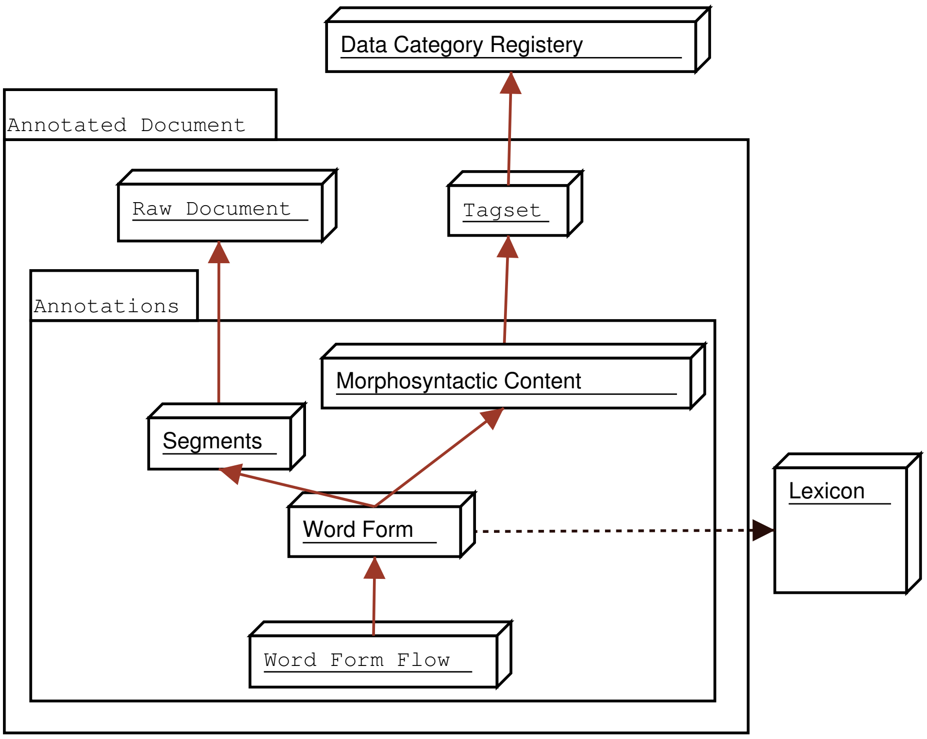

Figure 1: Simplified view of MSAF meta-model

Figure 1 presents a simplified view of the proposed meta-model for morpho-syntactic annotations. An annotated document is formed by a raw document and a set of annotations. The annotations are carried by *word forms* covering zero, one or more segments or `tokens` of the documents. The word forms are organized in one or more flows, because of ambiguities. These flows are materialized by *finite state automata*. A word form may reference a lexicon entry. The morpho-syntactic content attached to a word form is expressed by *feature structures* following the guidelines of one or more *tagsets*. The terminology or set of *categories* used in tagsets are described w.r.t. *registered data categories*. The current proposal addresses the representation of segments (through tokens), word forms, word form flows, morpho-syntactic content, and tagsets.

2 Structuring

Morpho-syntactic annotations preserve the linear characteristic of the «signifiant» (i.e. the annotated document), either in space or time, depending on the nature of the annotated document. These documents correspond to linear sequences, be they texts, vocal records, transcriptions, ...

Therefore, the sequences that may be decorated by a morpho-syntactic annotation are not necessarily characterized by character strings but more generally by some *span* (or *interval*) in various kinds of documents. This dissociation of the morpho-syntactic annotations from the original document may be also useful when these annotations conflict with some already present structuring of the document or when other kinds of annotation are to be attached that do not follow the same segmentation.

The *Text Encoding Initiative* (TEI) provides various mechanisms to identify sequences within documents and to link these sequences, as described in section 14.3 (*Blocks, Segments and Anchors*).

The current proposal for normalizing morpho-syntactic annotations should complete these mechanisms. In particular, it tries not to raise contradictions with TEI, in order to be able to mix TEI annotations and morpho-syntactic annotations. More generally, the possibility to easily add other levels of annotations (with their own segmentation) should be taken into account.

For instance, the segmentation of a document addresses positions in the original document that may be modified (insertion, displacement, deletion, ...). *TEI* (*14.1.1 Pointers and Links*) provides such a mechanism used to identify hypertextual links and other cross references with a document:

> `<ptr>` defines a pointer to another location in the current document in terms of one or more identifiable elements. target specifies the destination of the pointer by supplying the values used on the id attribute of one or more other elements in the current document

In order to reuse existing standards, such a mechanism will be used when possible to delimit linear sequences within textual documents. However, other kinds of position addressing will be considered for other kinds of documents (such a vocal recording), based on other existing standards (such as *MPEG7*).

2.1 Segmenting with tokens

Morpho-syntactic annotations are carried by segments *in praesentia* in the document flow, but this does not imply that the resulting segmentation corresponds to a sequence of adjacent segments partitioning the original document. It is particularly important to distinguish the morpho-syntactic units from their realizations. Some parts of a document may carry no annotations (typographic marks, didascalies, markup elements, ...); other parts may not exactly correspond to their segmented form (abbreviations, brachygraphies, typographic errors and variations, typographic and morphological contractions, ...). Also, a morpho-syntactic unit may not correspond to a segment identified by typographic marks (such as white spaces or hyphens), for instance for German compound words, speech transcription, or Sanskrit writing.

The element `token` is used to represent these segments of the original document that, roughly speaking, follow typographical, morphological, or phonological boundaries. The current proposal does not precise the linguistic properties of tokens. Among languages, a token may be identified through typographic properties (white-space, hyphens, characters, ...) and/or morphological properties (radical, affix, morpheme, ...). In any case, the description of the morphological, phonological or lexicologic structures that may define a token is not covered by the current proposal.

Other typographical marks used to format pages or to separate words and paragraphs, as well as encoding information, do not belong to morpho-syntactic annotations and are also not covered by this proposal, but rather by TEI.

In other words and as already mentioned, the element **token** provides an independence from the original document w.r.t. these aspects by providing a way to reference intervals in documents. The use of the attributes **from** and **to** are used to define such intervals. The content of these attributes depends on some chosen addressing schema. The following example shows an addressing schema based on the use of TEI element **ptr** (TEI Section 14.1.1 «Pointers and Links»):

```
<s><ptr target="p1"/>The <ptr target="p2"/>victim
<ptr target="p3"/>’<ptr target="p4"/>s
<ptr target="p5"/>friends <ptr target="p6"/>told
<ptr target="p7"/>police <ptr target="p8"/>that
<ptr target="p9"/>Krueger <ptr target="p10"/>drove
<ptr target="p11"/>into <ptr target="p12"/>the
<ptr target="p13"/>quarry <ptr target="p14"/>and
<ptr target="p15"/>never <ptr target="p16"/>surfaced
<ptr target="p17"/>.<ptr target="p18"/></s>
```

```
<token id="t1" from="p1" to="p2"/>
<token id="t2" from="p2" to="p4"/>
<token id="t3" from="p4" to="p5"/>
<token id="t4" from="p5" to="p6"/>
...
```

It is not always necessary to separate the original document from its annotations. For simple cases, when one wishes to directly enrich a document with morpho-syntactic annotations, an alternate notation for **token** is provided that can embed textual content.

```
<token id="t1">The</token>
<token id="t2">victim</token>
<token id="t3">’s</token>
<token id="t4">friends</token>
<token id="t5">told</token>
<token id="t6">police</token>
<token id="t7">that</token>
<token id="t8">Krueger</token>
<token id="t9">drove</token>
<token id="t10">into</token>
<token id="t11">the</token>
<token id="t12">quarry</token>
<token id="t13">and</token>
<token id="t14">never</token>
<token id="t15">surfaced</token>
<token id="t16">.</token>
```

This alternate notation can only be used when the morpho-syntactic annotations do no conflict with other kinds of annotations. Furthermore, the content of the textual material separating the textual content embedded within **token** is not precisely defined (white-space, newlines, no space, hyphen, ...).

QUESTION: Should we add some way to precise this inter token material, for instance with an attribute **glue** on **token**.

Two tokens may overlap on some part of the original document, for instance to denote some agglutinated or contracted form (such as, in French, «des» as a contraction for «de les» [of the]), or to denote multi-locutor documents with overlapping. In these cases, a `token` does not mark the realization of a typographical or vocal sequence, but expresses a deeper linguistic reality pertinent for segmenting a document.

Tokens address segments of the original document but also provide a level of possible abstraction w.r.t. this document, for instance w.r.t. graphical or phonological variations that are not linguistically pertinent. The non mandatory attribute `value` can be used to perform this abstraction, providing, for instance, the phonetic transcription of a speech segment, the roman transliteration of some Cyrillic word, the extension of an abbreviation, the corrected form of a words with typos, or the choice of a normalized form in presence of variations:

```
<token value="et␣caetera" id="t1">etc.</token>
<token value="tzar" id="t2">csar</token>
<token value="tzar" id="t3">tsar</token>
<token value="23/02/03" id="t4">February, 23rd 2003</token>
```

META: Add an example of phonetic transcription

The attribute `value` holds the linguistic interpretation of the content of a segment, pertinent to anchor morpho-syntactic annotations. Of course, this interpretation may differ from an user (or tool) to another one.

The abstraction provided by the attribute `value` is also adequate to handle the phenomena of contraction and agglutination where two tokens may cover the same segment of the original document for distinct values.

For instance, the following example illustrates the contraction of an abbreviation with a punctuation mark for «etc.», for the two variants of notations for element `token`:

a

```
<token value="et␣caetera" id="t1" from="p1" to="p3"/>
<token value="#dot#" id="t2" from="p1" to="p3"/>
```

b

```
<token value="et␣caetera" id="t1">etc.</token>
<token value="#dot#" id="t2"/>
```

META: Should we consider a reference notation such as

```
<token value="et␣caetera" id="t1">etc.</token>
<token value="#dot#" id="t2" idref="t1"/>
```

Another example, in Modern Greek, is provided by the idiomatic expression "καλόκαγαθος" (*good and brave*) that may be segmented in three agglutinated segments "καλός", "και", and "αγαθος" and represented by:

```
<token value="καλός" id="t0">καλο</token>
<token value="και" id="t1">κ</token>
<token value="αγαθός" id="t2">αγαθος</token>
```

QUESTION: The representation of overlapping tokens with the embedding notation is slightly confusing (see "etc." example), because the second token (e.g. '#dot#') seems to have no content, which is not true. Maybe a solution would be for the second token to refer the first one.

2.1.1 Formal description of `token`

```
tok =
  element token {
    attribute id { xsd:ID }?,
    attribute value { xsd:NCName }?,
    ( attribute from { DocumentLocation },
      attribute to { DocumentLocation }
    |
      text
    )
  }
```

The information `id`, `from`, `to`, `value` are implemented as attributes of element `token`. Content of attribute `id` should be an unique identifier with XML type `ID`.

2.2 Word Forms as linguistic units

The segment identified by `token` elements are used to anchor word forms, that may generally be associated to a lexical entry in an lexicon and also characterized by a part of speech as well as morphological and grammatical properties.

A token may be associated to more than one word form and, conversely, a word form may cover more than one token.

For instance, in French, the morphological agglutination of *auquel* («of whose») may have several representations, depending of the properties of the segmentation:

A. The character sequence *auquel* is not decomposed and is covered by a single `token`, with two word forms covering this segment.

```
<token value="auquel" id="t0">auquel</token>
<wordForm entry="à" tag="pos@prep" tokens="t0"/>
<wordForm entry="lequel" tag="pos@pronrel" tokens="t0"/>
```

B. the segmentation identifies two agglutinated parts materialized by two tokens, each of them anchoring a word form:

```
<token value="à" id="t0">auquel</token>
<token value="lequel" id="t1"/>
<wordForm entry="à" tag="pos@prep" tokens="t0"/>
<wordForm entry="lequel" tag="pos@pronrel" tokens="t1"/>
```

QUESTION: Should we add the other interpretations with poly-categories

These various interpretations can be motivated by the usage or by the available tools for a given language. All of them can be described following the current recommendation.

As mentioned before, there is no linguistic mandatory properties defining the tokens, which can for instance by automatically recognized by regular languages. On the other hand, a word form, that may cover zero, one or more tokens, should represent a linguistic unit that may carry morpho-syntactic information.

The current proposal does not discuss the linguistic choices that define these linguistic units but provide enough flexibility to annotate them. The choice may be motivated by lexical or morphological properties based on context and language (depending on the *nature* and *function* of words).

2.2.1 Token attachment

The simplest case of relationship between tokens and word forms is when a word form covers a single token.

```
<token id="t0" value="apple">apple</token>
<wordForm entry="apple" tokens="t0"/>
```

However, the current proposal allows the handling of more complex cases, as the identification of compound words covering several adjacent tokens:

```
<token id="t0" value="prime">prime</token>
<token id="t1" value="minister">minister</token>
<wordForm entry="prime_minister" tokens="t0␣t1"/>
```

A sequence of non connected tokens may also be attached to a word form, for instance to handle cases where some material is inserted inside the components of a word form:

```
<token value="afin" id="t1">afin</token>
<token value="justement" id="t2">justement</token>
<token value="de" id="t3">de</token>

<wordForm entry="afin_de" tokens="t1␣t3"/>
<wordForm entry="justement" tokens="t2"/>
```

This kind of phenomena may also occur for verbs with detached particles, for instance in English or German. The English infinitive verbal form "to <verb>" may also fit in this scheme.

```
<token value="to" id="t1">to</token>
<token value="eventually" id="t2">eventually</token>
<token value="decide" id="t3">decide</token>

<wordForm entry="decide" tokens="t1␣t3"/>
<wordForm entry="eventually" tokens="t2"/>
```

In order to identify discontinuous word-form while preserving some information about the position of each component in the flow of word forms, one may use word forms covering the same sequence tokens and referring to the same entry (but possibly sub-entries).

```
<token value="to" id="t1">to</token>
<token value="eventually" id="t2">eventually</token>
<token value="decide" id="t3">decide</token>

<wordForm entry="lexicon:decide:to" tokens="t1␣t3"/>
<wordForm entry="eventually" tokens="t2"/>
<wordForm entry="lexicon:decide:main" tokens="t1␣t3"/>
```

QUESTION: Should be add something more explicit to signal the fact the two word forms are actually strongly related, for instance an ID/IDREF mechanism on `wordForm`.

META: The notion of sub-entry is to be precised in the next subsection

Another case that may arise is when one wishes to materialize a word form without any realization in the original document and, therefore, associated with an empty sequence of tokens. It may be for instance the case of some pronouns in Spanish or the hypothesis of traces.

```
<token value="Jean" id="t1">Jean</token>
<token value="propose" id="t2">propose</token>
<token value="de" id="t3">de</token>
<token value="partir" id="t4">partir</token>

<wordForm entry="Jean" tokens="t1"/>
<wordForm entry="propose" tokens="t2"/>
<wordForm entry="de" tokens="t3"/>
<wordForm entry="PRO"/>
<wordForm entry="partir" tokens="t4"/>
```

It should be noted that a word form is associated to some possibly empty sequence of tokens but that it is positioned relatively to other word forms and not tokens, as will be detailed in Section 2.3. It implies that there is no problem with word forms that are not associated with tokens.

Finally, several word forms may be attached to a same token:

```
<!-- (Donne-le moi) -->
<token value="damelo" id="t1">Damelo</token>
<wordForm entry="da" tokens="t1"/>  <!-- (Donne) -->
<wordForm entry="me" tokens="t1"/>  <!-- (le) -->
<wordForm entry="lo" tokens="t1"/>  <!-- (moi) -->
```

2.2.2 Relationship to lexicons

META: Relationships to lexicons and limits: word form with entries, compound word forms

A word form is a linguistic unit identified by its morpho-syntactic properties. Generally, this linguistic unit may be characterized by a label corresponding to an entry if some lexicon. This identification is materialized by the attribute `entry`, whose content should express a reference to the lexicon entry.

```
  <token value="prime" id="t1">Prime</token>
  <token value="minister" id="t2">minister</token>
  <wordForm entry="lexicon:prime_minister" tokens="t1␣t2"/>
```

The notion of lexicon entry is outside the scope of this proposal. A reference to a lexicon entry is therefore not precisely defined and, in first approximation, should correspond to some XPath pointer. In particular, one may wish reference to “sub-entries” in lexicons, for polysemous entries or for compound forms:

```
<token value="to" id="t1">to</token>
<token value="eventually" id="t2">eventually</token>
<token value="decide" id="t3">decide</token>

<wordForm entry="lexicon:decide:to" tokens="t1␣t3"/>
<wordForm entry="eventually" tokens="t2"/>
<wordForm entry="lexicon:decide:main" tokens="t1␣t3"/>
```

META: The notation for referring lexicon entries should be precised and used in examples (XPath notation ?)

However, a sequence of tokens may sometimes be identified as forming a word form because of various properties but can not associated to some lexicon entry, either becase no lexicon is available or because the word form corresponds to a named entities (proper nouns, dates, ...) or to a neologism. In that case, the content of attribute `entry` may be left empty of filled by some informative label, that is not a reference:

```
<token id="t0">123</token>
<wordForm entry="NUMBER" tokens="t0"/>
```

The structure of compound forms may be expressed using nested word forms, therefore providing information about the subparts even when none is available for the whole, for instance for neologisms:

```
<!-- birthday gift wrapping paper -->
<token value="Geburtstags" id="t0">Geburtstags</token>
<token value="Geschenk" id="t1">geschenk</token>
<token value="Papier" id="t2">papier</token>
<wordForm tokens="t1 t2 t3">
   <wordForm entry="germanlex:geburstag" tokens="t1"/>
   <wordForm entry="germanlex:geschenk" tokens="t2"/>
   <wordForm entry="germanlex:papier" tokens="t3"/>
</wordForm>
```

2.2.3 Morpho-Syntactic content

At this point, a word form has been presented as covering a (possibly non connected) sequence of tokens, (possibly) referring to some lexicon entry. A word form should actually be completed by morpho-syntactic information precising its linguistic nature and its grammatical function in its current context.

More precisely, a *word form* carries a (complex) structure expressed by a *Feature Structure* that may attach one or more (possibly complex) values to a grammatical category or to some other linguistic property (i.e. part of speech, lemma, lexeme, morphology, ...).

META: Exact reference of FSR

This structure may also provides information of interest about morpho-syntax but not characterizing a word form, such as the origin of the information (a specific lexicon, a morphological tool, ...) or a confidence (returned by a stochastic tagger).

```
<token  id="t0" value="belle">belle</token>
<wordForm entry="lexicon:beau" tokens="t0">
<fs>
  <f name="cat"><sym value="adj"/></f>
  <f name="adj_type"><sym value="qual"/></f>
  <f name="gender"><sym value="fem"/></f>
  <f name="num"><sym value="sing"/></f>
</fs>
</wordForm>
```

Based on the FSR proposition, compact representation of feature structures can be used to fill the attribute `tag` of element `wordForm`.

```
<token  id="t0" value="belle">belle</token>
```

```
<wordForm entry="lexicon:beau" tag="cat@adj␣adj_type@qual␣gender@fem␣
    num@sing" tokens="t0"/>
```

This alternate notation will be sufficient for most cases of morpho-syntactic annotations, associated with the definition of a *tagset*.

The notion of tagset and the expressive power of Feature Structure is detailed in Section 3.

2.2.4 Formal description: `wordForm`

```
wordForm =
  element wordForm {
    attribute entry { xsd:NMTOKEN },
    ( attribute tag { string } | fs ),
    ( attribute tokens  { xsd:IDREFS } | tok+ ),
    wordForm*
  }
```

2.3 Handling ambiguities

The grouping of tokens into word forms as well as the lexical entry attached to a word form or its attached morpho-syntactic information may be ambiguous. It is therefore necessary to be able to handle these different cases of ambiguities.

A general and very generic answer is to materialize the possible readings as possible pathes through an Directed Acyclic Graph (DAG) whose edges are labeled by word form. Such DAGs forms a subpart of Finite State Automata and also cover the notion of *word lattice* used in the parsing and speech recognition communities. They can powerful enough to represent ambiguities between several decomposition into compound forms. They can also be used to denote simpler cases of lexical ambiguities.

For instance, the French textual sequence "*fer à cheval*" (horse shoe) can still be decomposed into several readings («[horse shoe]», «[iron] [on horse]», «[iron] [of] [horse]»), giving the following DAG:

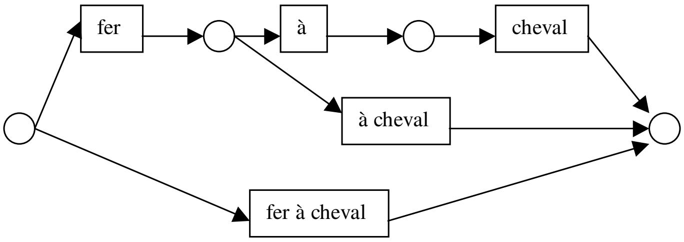

Figure 2: DAG de fer à cheval

```
<token value="fer" id="t1">fer</token>
<token value="à" id="t2">à</token>
<token value="cheval" id="t3">cheval</token>
<state id="S0" type="initial"/>
<state id="S2"/>
```

```
<state id="S3" type="final"/>
<transition source="S0" target ="S3">
   <wordForm entry="lexicon:fer_à_cheval" tokens="t1␣t2␣t3"/>
</transition >
<transition source="S0" target ="S1">
   <wordForm entry="lexicon:fer" tokens="t1"/>
</transition >
<transition source="S1" target ="S2">
   <wordForm entry="lexicon:à" tokens="t2"/>
</transition >
<transition source="S2" target ="S3">
   <wordForm entry="lexicon:cheval" tokens="t3"/>
</transition >
<transition source="S1" target ="S3">
   <wordForm entry="lexicon:à_cheval" tokens="t2␣t3"/>
</transition >
```

The linguistic units (`entry`) "fer à cheval", "fer", "à", "cheval" et "à cheval" correspond to minimal syntagmatic units that can be annotated.

Additional information could be added to edges such as probabilities.

2.3.1 Non ambiguous linear composition

When there is no ambiguity and one wishes to use simpler notation, the current proposal suggest a very simple alternate linear notation where element `wordForm` are implicitly chained following their appearance order, as illustrated by the following example:

```
<token value="fer" id="t1">fer</token>
<token value="à" id="t2">à</token>
<token value="cheval" id="t3">cheval</token>
<wordForm entry="fer" tokens="t1"/>
<wordForm entry="à" tokens="t2"/>
<wordForm entry="cheval" tokens="t3"/>
```

This notation would be a shortcut for the following canonical version:

```
<fsm init="s0" final="s3">
  <token id="t0" value="fer">fer</token>
  <token id="t1" value="à">à</token>
  <token id="t2" value="cheval">cheval</token>
  <state id="s0"/>
  <state id="s3"/>
  <transition source="s0" target="s3">
      <wordForm entry="lexicon:fer_à_cheval" tag="cat@noun␣..." tokens="
         t0␣t1␣t2"/>
  </transition>
  <state id="s1"/>
  <transition source="s0" target="s1">
       <wordForm entry="lexicon:fer" tag="cat@noun␣..." tokens="t0"/>
  </transition>
  <transition source="s1" target="s3">
       <wordForm entry="lexicon:à_cheval" tag="cat@adv" tokens="t1␣t2"/>
  </transition>
  <state id="s2"/>
  <transition source="s1" target="s2">
       <wordForm entry="lexicon:à" tag="cat@prep" tokens="t1"/>
  </transition>
  <transition source="s2" target="s3">
       <wordForm entry="lexicon:cheval" tag="cat@noun␣..." tokens="t2"/>
  </transition>
</fsm>
```

2.3.2 Lexical Ambiguities

Ambiguities between different lexical entries for a same sequence of tokens can be handled by the generic element **alt**:

```
<token  value="porte" id="t0">porte</token>
<alt>
  <wordForm tokens="t0" entry="lexicon:porte"  tag="cat@noun␣..."/>
  <wordForm tokens="t0" entry="lexicon:porter" tag="cat@verb␣..."/>
</alt>
```

Such a form is actually a shortcut for the expanded notation:

```
<token  id="t0" value="porte">porte</token>
<fsm init="s0" final="s1">
<state id="s0"/>
<state id="s1"/>
<transition source="s0" target="s1">
  <wordForm tokens="t0" entry="lexicon:porte"  tag="cat@noun␣..."/>
</transition>
<transition source="s0" target="s1">
  <wordForm tokens="t0" entry="lexicon:porter" tag="cat@verb␣..."/>
</transition>
</fsm>
```

2.3.3 Word form Content Ambiguities

The proposal on Feature Structure Representation provides several ways to represent ambiguities, for instance at the level of feature values. These mechanisms may be imported to avoid the use to DAG for ambiguities within the morpho-syntactic content of a word-form.

For instance, French verb form “mange” (to eat) is ambiguous between 1st and 3rd persons, which can be represented by the **vAlt** element proposed in FSR.

```
  <token  id="t0" value="mange">mange</token>
  <wordForm tokens="t0" entry="lexicon:manger">
     <fs>
       <f name="cat"><sym value="verb"/></f>
       <f name="aux"><sym value="avoir"/></f>
       <f name="mode"><sym value="indicatif"/></f>
       <f name="tense"><sym value="present"/></f>
       <f name="pers">
            <vAlt>
               <sym value="1"/>
               <sym value="3"/>
            </vAlt>
        </f>
        <f name="num"><sym value="sing"/></f>
      </fs>
  </wordForm>
```

A compact notation can still be used by registering most frequent cases of ambiguities in Feature Structure Libraries (see section about Tagsets).

```
<token  id="t0" value="mange">mange</token>
<wordForm tokens="t0" entry="lexicon:manger" tag="cat@verb␣aux@avoir␣
    mode@ind␣tense@pres␣pers@1|3␣num@sing"/>
```

2.3.4 Mixing simple and complex cases

Ambiguities are generally localized and it is tempting to also localize the use of the graph notation only where it is needed. The current proposal provides a way to switch between linear and graph notations.

```
<token  id="t0" value="afin">afin</token>
<token id="t1" value="de">de</token>
<fsm init="s0" final="s2">
  <transition source="s0" target="s2">
     <wordForm entry="lexicon:afin_de" tag="cat@prep" tokens="t0␣t1"/>
  </transition>
  <transition source="s0" target="s1">
     <wordForm entry="lexicon:afin" tag="cat@prep" tokens="t0"/>
  </transition>
  <transition source="s1" target="s2">
     <wordForm entry="lexicon:de" tag="cat@prep" tokens="t1"/>
  </transition>
</fsm>
<token id="t2" value="grandir">grandir</token>
<wordForm entry="lexicon:grandir" tag="cat@verb␣..." tokens="t2"/>
<token id="t3" value=",">,</token>
<wordForm entry="lexicon:," tag="cat@ponct" tokens="t3"/>
<token id="t4" value="il">il</token>
<wordForm entry="lexicon:il" tag="cat@pronoun␣..." tokens="t4"/>
<token id="t5" value="mange">mange</token>
<wordForm entry="lexicon:manger" tag="cat@verb␣..." tokens="t5"/>
<token id="t6" value="des(1:2)">des</token>
<wordForm entry="des(1:2)" tag="cat@adv" tokens="t6"/>
<token id="t7" value="des(2:2)"/>
<wordForm entry="lexicon:le" tag="cat@det␣..." tokens="t7"/>
<token id="t8" value="pommes">pommes</token>
<token id="t9" value="de">de</token>
<token id="t10" value="terre">terre</token>
<fsm init="s8" final="s11">
  <transition source="s8" target="s11">
     <wordForm entry="lexicon:pomme_de_terre" tag="cat@noun␣..." tokens=
         "t8␣t9␣t10"/>
  </transition>
  <transition source="s8" target="s9">
```

```
    <wordForm entry="lexicon:pomme" tag="cat@noun␣..." tokens="t8"/>
  </transition>
  <transition source="s9" target="s10">
    <wordForm entry="lexicon:de" tag="cat@prep" tokens="t9"/>
  </transition>
  <transition source="s10" target="s11">
    <wordForm entry="lexicon:terre" tag="cat@noun␣..." tokens="t10"/>
  </transition>
</fsm>
```

A shortcut notation for the rather long expanded version:

```
<fsm init="s0" final="s11">
  <token id="t0" value="afin">afin</token>
  <token id="t1" value="de">de</token>
  <state id="s0"/>
  <state id="s2"/>
  <transition source="s0" target="s2">
    <wordForm entry="lexicon:afin_de" tag="cat@prep" tokens="t0␣t1"/>
  </transition>
  <state id="s1"/>
  <transition source="s0" target="s1">
    <wordForm entry="lexicon:afin" tag="cat@prep" tokens="t0"/>
  </transition>
  <transition source="s1" target="s2">
    <wordForm entry="lexicon:de" tag="cat@prep" tokens="t1"/>
  </transition>
  <token id="t2" value="grandir">grandir</token>
  <state id="s3"/>
  <transition source="s2" target="s3">
    <wordForm entry="lexicon:grandir" tag="cat@verb␣..." tokens="t2"/>
  </transition>
  <token id="t3" value=",">,</token>
  <state id="s4"/>
  <transition source="s3" target="s4">
   <wordForm entry="lexicon:," tag="cat@ponct" tokens="t3"/>
  </transition>
  <token id="t4" value="il">il</token>
  <state id="s5"/>
  <transition source="s4" target="s5">
   <wordForm entry="lexicon:il" tag="cat@pronoun␣..." tokens="t4"/>
  </transition>
  <token id="t5" value="mange">mange</token>
  <state id="s6"/>
  <transition source="s5" target="s6">
   <wordForm entry="lexicon:manger" tag="cat@verb␣..." tokens="t5"/>
  </transition>
  <token id="t6" value="des(1:2)">des</token>
  <state id="s7"/>
  <transition source="s6" target="s7">
   <wordForm entry="des(1:2)" tag="cat@adv" tokens="t6"/>
  </transition>
  <token id="t7" value="des(2:2)"/>
  <state id="s8"/>
  <transition source="s7" target="s8">
    <wordForm entry="lexicon:le" tag="cat@det␣..." tokens="t7"/>
  </transition>
  <token id="t8" value="pommes">pommes</token>
  <token id="t9" value="de">de</token>
  <token id="t10" value="terre">terre</token>
  <state id="s11"/>
  <transition source="s8" target="s11">
    <wordForm entry="lexicon:pomme_de_terre" tag="cat@noun␣..." tokens="
        t8␣t9␣t10"/>
  </transition>
  <state id="s9"/>
  <transition source="s8" target="s9">
    <wordForm entry="lexicon:pomme" tag="cat@noun␣..." tokens="t8"/>
  </transition>
  <state id="s10"/>
  <transition source="s9" target="s10">
    <wordForm entry="lexicon:de" tag="cat@prep" tokens="t9"/>
  </transition>
  <transition source="s10" target="s11">
    <wordForm entry="lexicon:terre" tag="cat@noun␣..." tokens="t10"/>
  </transition>
</fsm>
```

2.4 Formal description: `fsm`, `state`, `transition`

```
fsm =
  element fsm {
  attribute init {xsd:IDREF}?,
  attribute final {xsd:IDREF}?,
  (tok | state | transition )+
  }
state =
   element state {
         attribute id { xsd:ID }
   }
transition =
   element transition {
      attribute source { xsd:IDREF },
      attribute target { xsd:IDREF },
      (wordForm | wordFormAlt)
   }
wordFormAlt =
  element alt { wordForm+ }
```

Also a question about alternatives on word forms: should we use a generic element `alt` or define a specialized version `wordFormAlt`.

3 Morpho-Syntactic annotation content

The previous section explain how to complete a document with morpho-syntactic annotations. However, it does not precise the content of these annotations. What set of features and feature values to use to express this content (within element `wordForm`) and with which meaning ?

Such a set is usually referred as a *tagset* specifying the content of possible annotations. However, the diversity of approaches and languages makes almost impossible the proposition of an unique tagset. More modestly or pragmatically, the current proposal is to provide mechanisms to define tagsets by relying on a Data Category Registry (DCR) and Feature Structures (FSR).

An annotated document will therefore be completed by either adding or referring to a *tagset*.

3.1 Representing Feature Structures

This section illustrates, trough a few examples, how the FSR proposal may be used to represent morpho-syntactic content. A morpho-syntactic content attached to a word form may be represented by a set of pairs (attribute, value) called *feature specifications*. Following FSR, such a simple set is represented by:

```
<wordForm entry="prime␣minister" tokens="t1␣t2">
<fs>
     <f name="pos">
        <sym value="noun"/>
     </f>
     <f name="gender">
        <sym value="masculine"/>
     </f>
     <f name="number">
        <sym value="singular"/>
     </f>
</fs>
</wordForm>
```

3.1.1 Disjunctive values

FSR provides way to denotes alternatives for values.

```
<token  value="mange" id="t0">mange</token>
<wordForm entry="lexicon:manger" tokens="t0">
  <fs>
    <f name="cat"><sym value="verb"/></f>
    <f name="aux"><sym value="avoir"/></f>
    <f name="mode"><sym value="indicatif"/></f>
    <f name="tense"><sym value="present"/></f>
    <f name="pers">
      <vAlt>
        <sym value="1"/>
        <sym value="3"/>
      </vAlt>
    </f>
    <f name="num"><sym value="sing"/></f>
  </fs>
</wordForm>
```

3.1.2 Multiple Tags

One may wish to assign a sequence of tags to a word form, for instance for some contracted form. A way to do that is to use the multiple values `vMult` provided by FSR.

```
<token id="t0" value="auxquels">auxquels</token>
<wordForm entry="lexicon:auquel" tokens="t0">
  <vMult org="list">
      <fs>
         <f name="cat"><sym value="prep"/></f>
      </fs>
      <fs>
         <f name="cat"><sym value="pronoun"/></f>
         <f name="kind"><sym value="rel"/></f>
         <f name="num"><sym value="pl"/></f>
         <f name="gender"><sym value="masc"/></f>
      </fs>
  </vMult>
</wordForm>
```

Note: In general, an alternate solution for contracted word is to set up several word forms attached to a same token, each of them with a simple tag, even if each word form is some subpart of a lexical entry.

```
<token id="t0" value="auxquels">auxquels</token>
<wordForm entry="lexicon:auquel/à" tokens="t0">
    <fs>
         <f name="cat"><sym value="prep"/></f>
    </fs>
</wordForm>
<wordForm  entry="lexicon:auquel/quel" tokens="t0">
     <fs>
         <f name="cat"><sym value="pronoun"/></f>
         <f name="kind"><sym value="rel"/></f>
         <f name="num"><sym value="pl"/></f>
         <f name="gender"><sym value="masc"/></f>
     </fs>
</wordForm>
```

3.2 Compact notations for morpho-syntactic annotations

FSR proposal provides ways for compact representations of feature structure, essentially by defining *libraries* of feature structures, feature specifications, and feature values. These mechanisms may be used to fill the content of attribute `tag` for element `wordForm` by compact tags, following a standard practice in the NLP community.

The generic way provided by FSR to use libraries is illustrated by the following example, with the attribute `feats` of element `fs`:

```
<!-- A feature value library -->
<vLib>
     <sym id="noun" value="noun"/>
     <sym id="sing" value="singular"/>
     <sym id="masc" value="masculine"/>
</vLib>
<!-- A feature specification library -->
<fvLib>
   <f id="pos@n" name="pos" fVal="noun"/>
   <f id="num@sing" name="num" fVal="sing"/>
   <f id="gen@fem" name="gen" fVal="fem"/>
</fvLib>
<!-- Annotated document -->
<wordForm entry="prime_minister" tokens="t1">
      <fs feats ="pos@n␣num@sing␣gen@fem"/>
</wordForm>
```

The current proposal allows the use of attribute `tag` to assign a compact tag directly to a word form:

```
<wordForm entry="prime_minister" tokens="t1" tag="pos@n␣num@sing␣
    gen@masc"/>
```

Even disjunctive values may be simplified, following the same mechanism:

```
<!-- A feature value library -->
<vLib>
     <vAlt id="1|3">
          <sym value="1"/>
          <sym value="3"/>
     </vAlt>
      ...
</vLib>
<!-- A feature specification library -->
<fvLib>
   <f id="pers@1|3" name="pers" fVal="1|3"/>
    ...
</fvLib>
<!-- Annotated document -->
<token  id="t0" value="porte">porte</token>
<wordForm entry="lexicon:porter" tag="cat@verb␣pers@1|3␣num@sing␣..."
    tokens="t0"/>
```

3.3 Designing tagsets with Morpho-Syntactic Data Categories

META: This section should be completed.

The attributes, values, and possibly feature types used to specify morpho-syntactic content are not just labels but carry linguistic meanings, in other word a semantic. To avoid misinterpretations, this semantic should be clearly defined.

However, the current proposal does not try to define the semantic of an unique complete set of such attributes, values, and types. It would be almost impossible to be complete (given the diversity of languages) and almost impossible to assign to each element a semantic accepted by the whole community.

Instead, it is proposed that an annotated document should be completed by the indication of one or more *tagsets*.

The first objective of a tagset is to list the terminology used to annotate a document, precising its semantic w.r.t, an official registered terminology, namely a *Data Category Registery* for Morpho-Syntax. The process may be seen as selecting a set of data categories (*Data Category Selection – DCS*).

A private category may be introduced by selecting some registered one:

```
<dcs private="genre" registered="dcs:morphosyntax:gender:french" rel="
    eq"/>
<dcs private="fem" registered="
    dcs:morphosyntax:gender:french:femininin"/>
```

The correspondance with a registered category may not be perfect. The `rel` may be used to specify which relationship exists between the private and registered categories. For instance, one may introduce a private category advneg as being subsumed by a more general registered category adverb.

```
<dcs private="advneg" registered="dcs:morphosyntax:pos:adverb" rel="
    subs"/>
<dcs private="strange" rel="none"/>
```

It is also possible (but not advised) to introduce a private category bearing no relationship with any registered category.

```
<dcs private="title"/>
```

When the correspondance is not perfect, a few words of description should be added to precise the meaning of the private category.

```
<dcs private="title">
    <description> A part of speech used to denote honorific titles like
    Pr. or S.A.S.
    </description>
</dcs>
```

QUESTION: The proposed scheme only cover very elementary mappings. Should we propose more sophisticated mappings as equivalence between feature structures,

The second objective of an tagset is to specify the set of valid feature structures based on the introduced terminology. It will be achieved by relying on a ISO standard to come about Feature Structure Declaration.

A third objective of an tagset is to name the most common morphosyntactic structures through the use of feature structure libraries, in order to be able to use compact tags for annotations.

3.3.1 Formal description: `tagset`

META: TO BE DEFINED

```
tagset =
    element tagset {
          ( attribute ref { xsd:anyURL }
          | dcs*,fsd,fsLibs*
          )
      }
dcs =
      element dcs {
          attribute private { xsd:NCName },
          ( attribute registered { xsd:anyURL },
            attribute rel { "eq" | "subs" | "gen" } )?
          element description { text }?
      }
## fsLibs: to be imported
## fsd: to be imported (and defined)
```

The `dcs` corresponds to a Data Category Selection part whose exact content is still to be defined.

The `fsd` corresponds to a Feature Structure Declaration part whose normalization is yet to be done.

4 Global information

The current proposal does not address the issue of adding metadata, globally and on annotations, for instance to mention the date (creation, revision) of an annotation and its author(s).

A global element `msa` (for **Morpho-Syntactic Annotations**) is introduced as root element to contain morpho-syntactic annotations. This element `msa` is used to specify global properties relative to the annotated documents, in particular the addressing scheme used to delimit segments. One or more tagsets should also be mentioned either explicitly or by reference with an URL.

META: TO BE COMPLETED

```
<msa addressing="mpeg7">
  <token id="t0" from="" to="" value="">
␣␣<wordForm␣tokens="t0"␣entry="">␣...␣</wordForm>
␣␣...
</msa>
```

4.1 Formal description

```
start =
        element maf {
           addressing,
           tagset,
           (fsm | tok | wordForm | wordFormAlt )+
        }
## To be precised
addressing =
      attribute addressing { xsd:NMTOKEN }
DocumentLocation = xsd:NMTOKEN
```

META: The notion of *addressing scheme reference* is yet to be precised. A possible list of such schema would include:

- **TEI ptrs,**
- **XPath,**
- **byte offsets,**
- **Unicode char offsets,**
- **MPEG7 multimedia addressing**

A Informative Relax NG compact schema

```
# $Id$

# Preliminary Relax NG schema for MAF: Morpho Syntactic Annotations
# Eric de la Clergerie <Eric.De_La_Clergerie@inria.fr>

default namespace = ""

start =
    element maf {
        addressing,
        tagset,
        (fsm | tok | wordForm | wordFormAlt )+
    }
## To be precised
addressing =
    attribute addressing { xsd:NMTOKEN }
tok =
    element token {
        attribute id { xsd:ID }?,
        attribute value { xsd:NCName }?,
        ( attribute from { DocumentLocation },
            attribute to { DocumentLocation }
          |
            xsd:NMTOKEN
        )
    }
wordForm =
    element wordForm {
        attribute entry { xsd:NMTOKEN },
        ( attribute tag { string } | fs ),
        ( attribute tokens  { xsd:IDREFS } | tok+ ),
        wordForm*
    }
fsm =
    element fsm {
        attribute init {xsd:IDREF}?,
        attribute final {xsd:IDREF}?,
        (tok | state | transition )+
    }
state =
    element state {
        attribute id { xsd:ID }
    }
transition =
    element transition {
        attribute source { xsd:IDREF },
        attribute target { xsd:IDREF },
        (wordForm | wordFormAlt)
    }
wordFormAlt =
    element alt { wordForm+ }
```

```
tagset =
   element tagset {
      ( attribute ref { xsd:anyURL }
       | dcs*,fsd,fsLibs*
      )
   }
dcs =
   element dcs {
      attribute private { xsd:NCName },
      ( attribute registered { xsd:anyURL },
         attribute rel { "eq" | "subs" | "gen" } )?
      element description { text }?
   }
## To be precised
DocumentLocation = xsd:NMTOKEN
```

B Informative DTD

```
<?xml version="1.0" encoding="UTF-8"?>


    <!--

    DTD for Morpho-Syntaxic Annotation Framework

  -->

<!ENTITY % fs-SYSTEM 'INCLUDE'>
<!ENTITY % fs-PUBLIC 'IGNORE'>

<![%fs-SYSTEM[
        <!ENTITY % dtd-fs SYSTEM "fs.dtd">
]]>
<![%fs-PUBLIC[
        <!ENTITY % dtd-fs PUBLIC "-//DTD␣fs//DTD//EN" "http://www.tc37sc4.org/dtd/fs.dtd">
]]>
%dtd-fs;


  <!ELEMENT msa          ((token | wordForm | wordFormAlt | fsm)+) >

  <!ELEMENT token        (#PCDATA) >
  <!ATTLIST token        value CDATA #IMPLIED
                         from IDREF #IMPLIED
                         to IDREF #IMPLIED
                         id ID #IMPLIED>

  <!ELEMENT wordForm     (fs | wordForm | token)*>
  <!ATTLIST wordForm     tokens IDREFS #IMPLIED
                         entry CDATA #IMPLIED
```

```
                       tag CDATA #IMPLIED>


<!ELEMENT wordFormAlt  (wordForm+)>

<!ELEMENT fsm          ((state | transition | token)+)>
<!ATTLIST fsm          initial IDREF #REQUIRED
                       finals  IDREFS #REQUIRED>

<!ELEMENT state        EMPTY>
<!ATTLIST state        id ID #REQUIRED>

<!ELEMENT transition   (wordFormAlt | wordForm)>
<!ATTLIST transition   from IDREF #REQUIRED
                       to IDREF #REQUIRED>
```