TEI Feature Structures as a Representation Format for Multiple Annotation and Generic XML Documents

Jens Stegmann

Bielefeld University

Andreas Witt

Institute for the German Language, Mannheim

Outline

Mitglied der Leibniz Gemeinschaft

1. Introduction
2. Feature Structures
   - Feature Structures in a Nutshell
   - The TEI Tag Set for Feature Structures
3. Representation and Transformation
   - Representation of XML Documents via TEI FSs
   - Aspects of the XSLT Implementation
4. Summary and Outlook

Introduction

Representation format 'feature structures'

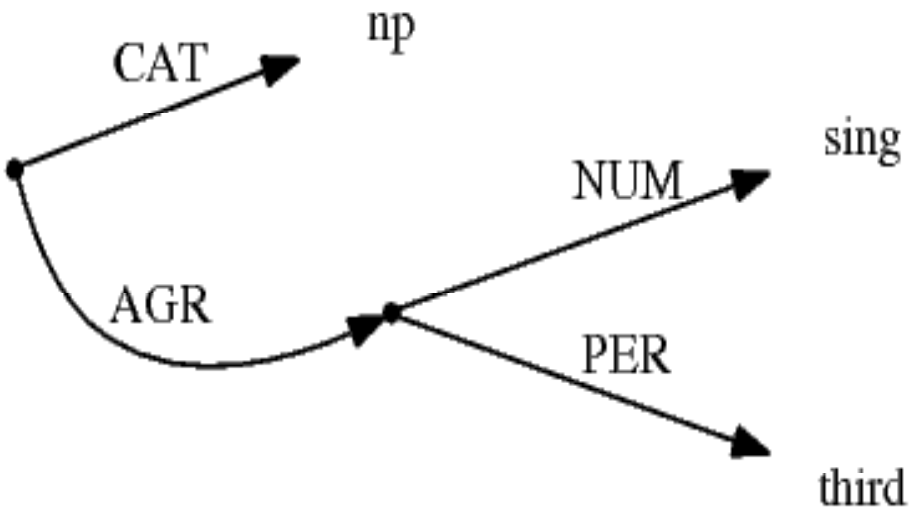
Mitglied der Leibniz Gemeinschaft

- in 2006 TEI's representation format for FS was adopted by ISO TC37 SC4 as a standard (ISO 24610)
- we propose to use feature structures as a representation format for annotated documents
- possible domains of application
  - Multiple Annotation: TEI FSs as an integrative representation format for sets of XML annotated documents
  - generic XML document instances: FS encoding may allow for the application of general operations defined on FSs as well as specific tools developed in computational linguistics

Example from the field of linguistics

Mitglied der Leibniz Gemeinschaft

- a feature structure for nominal phrases (3rd person, singular)
- graph representation



Attribute Value Matrix Notation

Mitglied der Leibniz Gemeinschaft

$$\begin{bmatrix} \text{CAT} & \text{np} \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sing} \\ \text{PER} & \text{third} \end{bmatrix} \end{bmatrix}$$

TEI/ISO-based XML-representation

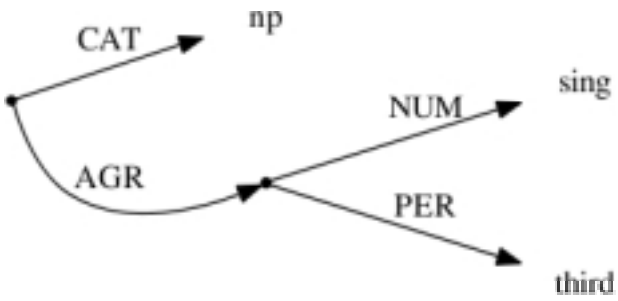
Mitglied der Leibniz Gemeinschaft

```
<fs>
    <f name="CAT"><symbol value="np" /></f>
    <f name="AGR">
        <fs><f name="NUM"><symbol value="sing" /></f>
            <f name="PER" /><symbol value="third" /></f>
        </fs>
    </f>
</fs>
```

The three display syntaxes together

Mitglied der Leibniz Gemeinschaft

```
<fs>
 <f name="CAT">
  <symbol value="np" />
 </f>
 <f name="AGR">
  <fs>
   <f name="NUM">
    <symbol value="sing" />
   </f>
   <f name="PER" />
    <symbol value="third" />
   </f>
  </fs>
 </f>
</fs>
```



$$\begin{bmatrix} \text{CAT} & \text{np} \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sing} \\ \text{PER} & \text{third} \end{bmatrix} \end{bmatrix}$$

Applications of FSs

Mitglied der Leibniz Gemeinschaft

- representation of linguistic information, wrt:
  - lexicon
  - grammar
  - semantics
  - …
- general knowledge representation format

  hence:

  we can use it to represent the information given in XML-documents

Application Scenario I:

Multiple Annotation

- Multiple Annotation allows for:
  - annotating 'incompatible' hierarchies, e.g. the annotation of the word "tables" wrt. phonology and morphology

    Anno1: <w><m>table</m><m>s</m></w>

    Anno2: <w><syll>ta</syll><syll>bles</syll></w>
  - dealing with heterogeneous tag sets:

    e.g. information units from phonology, morphology, syntax, semantics, and pragmatics

Multiple Annotation

Mitglied der Leibniz Gemeinschaft

- implementing redundant encoding in multiple forms: a separate XML document instance for each annotation level

  Doc1: <doc n="d1">…
  <w><m>table</m><m>s</m></w>…</doc>

  Doc2: <doc n="d1">…
  <w><syll>ta</syll><syll>bles</syll></w>…<doc>

- advantages:
  - all annotation layers are equally accessible and readable
  - extensible: new layers can be added at any time
  - annotation layers are reusable and interoperable without reference to other layers

FSs as an integrative representation

Mitglied der Leibniz Gemeinschaft

- transformation of multiply annotated text to an adequate integrative representation format
- TEI feature structures can serve as such an integrative representation format (thanks to Lou Burnard)
- we developed
  - FS-based format
  - transformations of the multiply annotated document to a single feature structure

Application Scenario II: Generic XML documents

Mitglied der Leibniz Gemeinschaft

- encoding 'normal' XML-documents by means of feature structures
- possible advantages:
  - tools of computational linguists may become applicable: bridging the gap of annotations and text content
  - feature structures might serve as knowledge representation format: allows for combining information contained in documents and representation of world knowledge
  - general operations (e.g. unification) are applicable
  - feature structures are well understood mathematical objects

Feature Structures

Feature Structures in a Nutshell

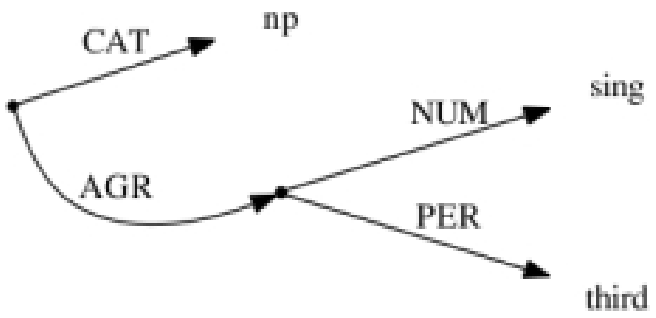
Mitglied der Leibniz Gemeinschaft

- general purpose data structure, knowledge representation
- mathematical grounding: graphs and partial functions
- partial function perspective: FS construed as a partial function from a set of features to a set of values
- values can be either atomic, e.g. symbolic or binary, or complex (e.g. a FS or list)

Graph perspective

Mitglied der Leibniz Gemeinschaft

- graphs are mathematical entities that consist of nodes and edges; the edges connect the nodes of a graph
- graphs can be depicted as graph display diagrams
- FSs are rooted labeled directed (acyclic) graphs
  - labeled edges represent the features
  - leaf nodes represent the atomic values
  - inner nodes represent complex values



Attribute value matrix notation

Mitglied der Leibniz Gemeinschaft

- attribute value matrices (AVMs) might be used as an alternative to the graph visualization
- values to the right of their corresponding features, brackets indicate the scope of (sub-)feature structure(s)
- N.B. some linguistic theories use different notations (graphs or AVMs) to distinguish "(total) models" and "(partial) descriptions"

$$\begin{bmatrix} \text{CAT} & \text{np} \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \text{sing} \\ \text{PER} & \text{third} \end{bmatrix} \end{bmatrix}$$

Partiality and underspecification

Mitglied der Leibniz Gemeinschaft

- feature structures list correct and only correct information, but not necessarily all the information wrt. the objects they describe, i.e. they may be partial
- partiality allows to capture generalizations via the underspecification of certain characteristics, e.g. the class of NPs vs. the class of NPs 3rd person singular

Structure sharing

Mitglied der Leibniz Gemeinschaft

- features may have identical values, this is represented by ‘structure sharing’
- features are associated with the same value token
- also known as: co-reference, reentrancy
- in AVM notation: indicated by co-indexed boxes (either referring to or indicating the shared value token)
- in graph notation: edges lead into the same node

Subsumption and unification

Mitglied der Leibniz Gemeinschaft

- subsumption: FS *F* subsumes a FS *H* if *F* contains a subset of the information in *H*
  - *F* carries less information than *H*, or
  - *F* is more general than *H*
- unification: the result of the unification of FSs *F* and *G*, if any, is the most general FS *H* such that *F* subsumes *H* and *G* subsumes *H*. If the feature structures are incompatible, the unification fails

Generalization

Mitglied der Leibniz Gemeinschaft

- the dual of unification
- generalization: the result of the generalization of two feature structures *F* and *G* is the most specific feature structure *E*, such that *E* subsumes *F* and *E* subsumes *G*.
- generalization can not fail: the result of the generalization of incompatible FSs is the empty FS [ ] , since [ ]subsumes every feature structure

TEI Tag Set for Feature Structures

Mitglied der Leibniz Gemeinschaft

- part of the TEI Guidelines: P3, P4, P5, and ISO 24610
- foundational elements:
  - <fs> for feature structures, and
  - <f> for features
- features: **name** attribute on <f> for the feature name, the element content expresses the desired value
  - for atomic feature values: <**binary**>, <**symbol**>, and <**numeric**> go with a **value** attribute, <**string**> uses its element content to express the desired value

Collections and structure sharing

Mitglied der Leibniz Gemeinschaft

- collection values: <vColl> elements with **org** attribute that specifies the collection type (**list**, **set**, or **bag**)
- the element content is a sequence of value representations
- structure sharing: <vLabel> with associated **name** attribute that gives the index (in AVM syntax); <vLabel> either contains the value token itself OR it is empty and makes reference to a token specified by the content of another <vLabel> element with the same **name** value

Example: AVM notation of structure sharing and collection values

Mitglied der Leibniz Gemeinschaft

$$\begin{bmatrix} \mathrm{F} & \left\langle \boxed{a}\begin{bmatrix} \mathrm{I} & \mathrm{a} \\ \mathrm{J} & \mathrm{b} \end{bmatrix}, \boxed{b}\begin{bmatrix} \mathrm{K} & \mathrm{c} \\ \mathrm{L} & \mathrm{d} \end{bmatrix} \right\rangle \\ \mathrm{G} & \boxed{a} \\ \mathrm{H} & \left\{ \boxed{b}, \begin{bmatrix} \mathrm{M} & \mathrm{e} \\ \mathrm{N} & \mathrm{f} \end{bmatrix} \right\} \end{bmatrix}$$

Example: TEI notation of structure sharing and collection values

Mitglied der Leibniz Gemeinschaft

```
<fs><f name="F">
  <vColl org="list"><vLabel name="a">
                    <fs><f name="I"><symbol value="a"/></f>
                        <f name="J"><symbol value="b"/></f></fs></vLabel>
                   <vLabel name="b">
                    <fs><f name="K"><symbol value="c"/></f>
                        <f name="L">
                          <symbol value="d"/></f></fs></vLabel></vColl></f>
<f name="G"><vLabel name="a"/></f>
<f name="H">
  <vColl org="set"><vLabel name="b"/>
                    <fs><f name="M"><symbol value="e"/></f>
                        <f name="N"><symbol
   value="f"/></f></fs></vColl></f></fs>
```

Representation and Transformation

Representing XML Docs with FS

Mitglied der Leibniz Gemeinschaft

- perspective of graph theory: XML documents are ordered trees, while FSs are unordered directed acyclic graphs
- because: possibility of structure sharing and no order among features on identical hierarchical levels in a FS
- representing generic XML documents as TEI FSs: mapping to a less rigid structure, which means that we need to find a way to conserve that information
- specifically: how shall we represent the sequential and hierarchical relations that hold within XML documents?

Representational matters

Mitglied der Leibniz Gemeinschaft

- idea:
  - we use specific features to express hierarchical relations
  - the associated values will themselves be structured and that structure can be interpreted as reflecting the sequential relations
- in the following: discussion of two representation alternatives that follow this approach

Example annotations

Mitglied der Leibniz Gemeinschaft

- the German verb "geben" ("to give"): morphology and phonology

```
<w>
    <m type="lexical">geb</m>
    <m type="flexive">en</m>
</w>

<w>
    <syll n="s1">ge</syll>
    <syll n="s2">ben</syll>
</w>
```

First representation alternative

Mitglied der Leibniz Gemeinschaft

- top-level features: DATA and TIER *1*, …, TIER*n*
- sequential relations are represented by recursively defined list structures: FIRST/REST-notation for sequences
- feature used for hierarchical relationships: CONTENT
- DATA's value is a representation of the common textual characters of the document that conforms to the FIRST/REST-scheme
- the content of the separate annotation documents is located within the numbered TIER features

Representation alternative I as TEI-FS

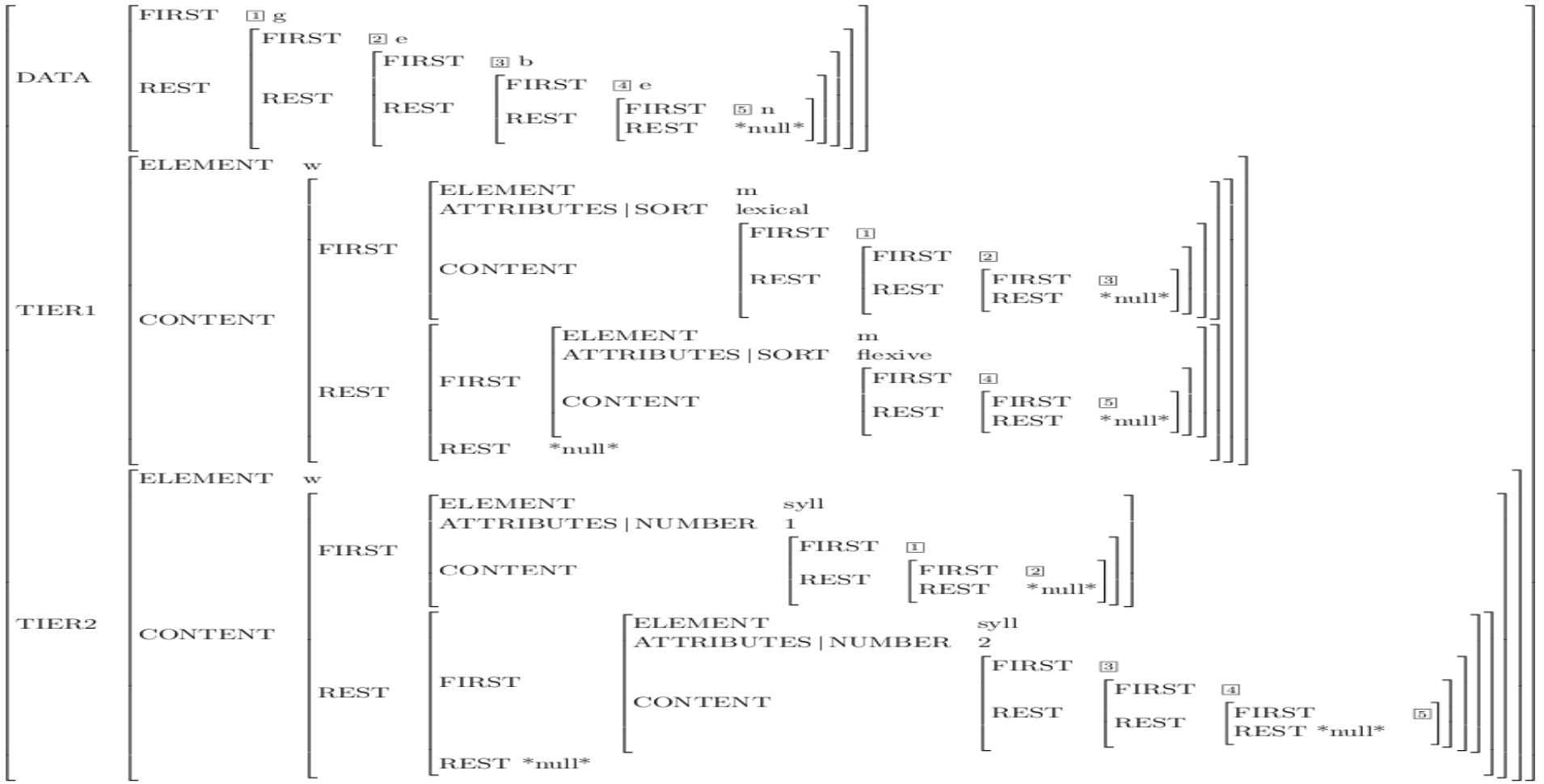
Mitglied der Leibniz Gemeinschaft

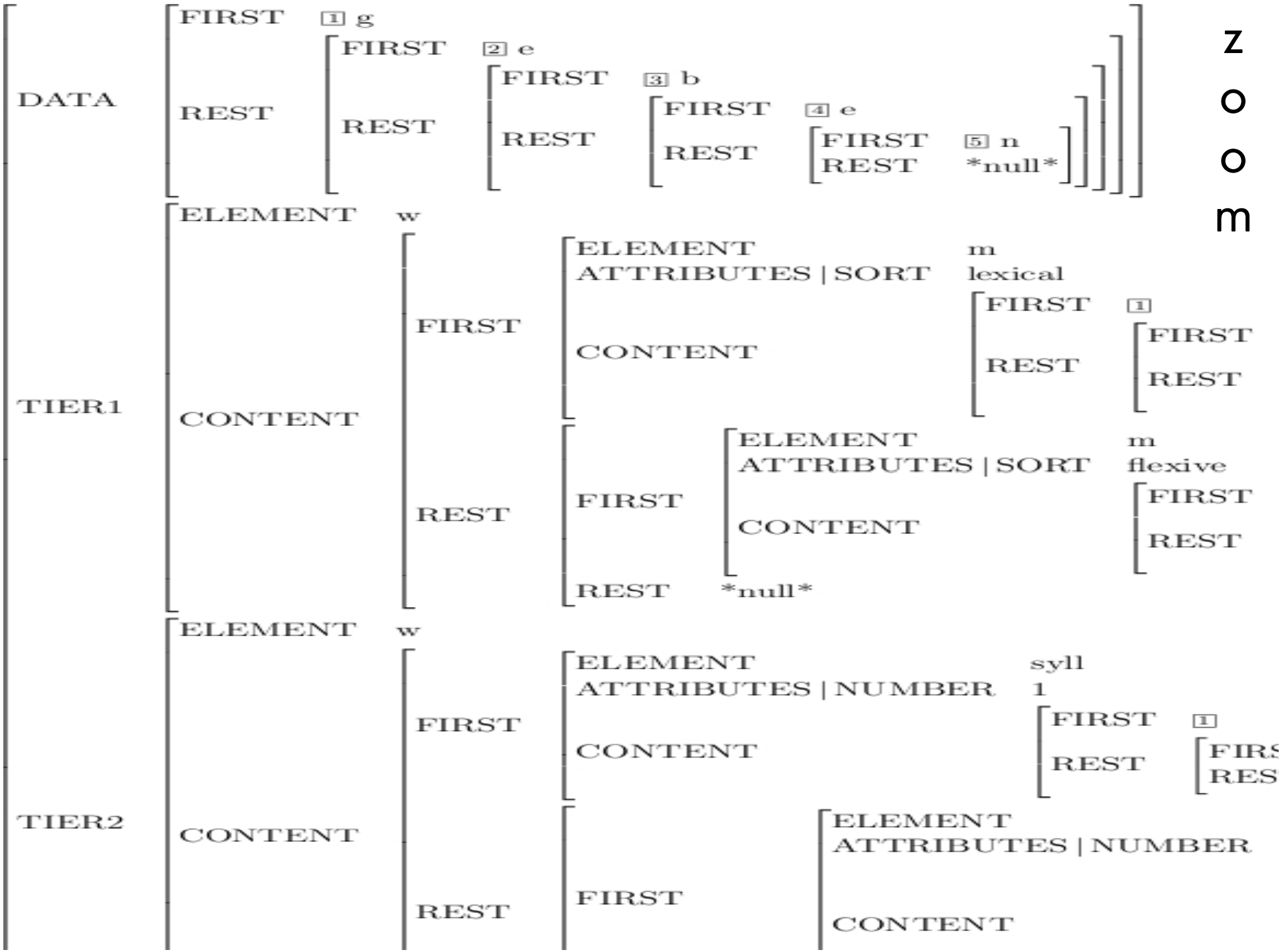
```
<fs> <f name="DATA">
    <fs><f name="FIRST"><vLabel name="1"><symbol value="g"/></vLabel></f>
        <f name="REST">
            <fs><f name="FIRST">
                    <vLabel name="2"><symbol value="e"/></vLabel></f>
                <f name="REST">
                    <fs><f name="FIRST">
                            <vLabel name="3"><symbol value="b"/></vLabel></f>
                <f name="REST">
                    <fs><f name="FIRST">
                            <vLabel name="4"><symbol value="e"/></vLabel></f>
                        <f name="REST">
                            <fs><f name="FIRST"><vLabel name="5">
                                    <symbol value="n"/></vLabel> </f>
                            <f name="REST"><symbol value="*null*"/> </f></fs>
                                    </f></fs></f></fs> </f> </fs>     </f>
  </fs> </f>
```

```
<f name="TIER1"> ... </f> <f name="TIER2"> ... </f> </fs>
```

Representation alternative I as AVM

Mitglied der Leibniz Gemeinschaft




DATA
FIRST
g
REST
FIRST
e
REST
FIRST
b
REST
FIRST
e
REST
FIRST
n
REST
null
TIER1
ELEMENT
w
CONTENT
FIRST
ELEMENT
m
ATTRIBUTES | SORT
lexical
CONTENT
FIRST
REST
FIRST
REST
REST
ELEMENT
m
ATTRIBUTES | SORT
flexive
CONTENT
FIRST
REST
REST
null
TIER2
ELEMENT
w
CONTENT
FIRST
ELEMENT
syll
ATTRIBUTES | NUMBER
1
CONTENT
FIRST
REST
REST
ELEMENT
ATTRIBUTES | NUMBER
CONTENT
z
o
o
m

Representation alternative 2

Mitglied der Leibniz Gemeinschaft

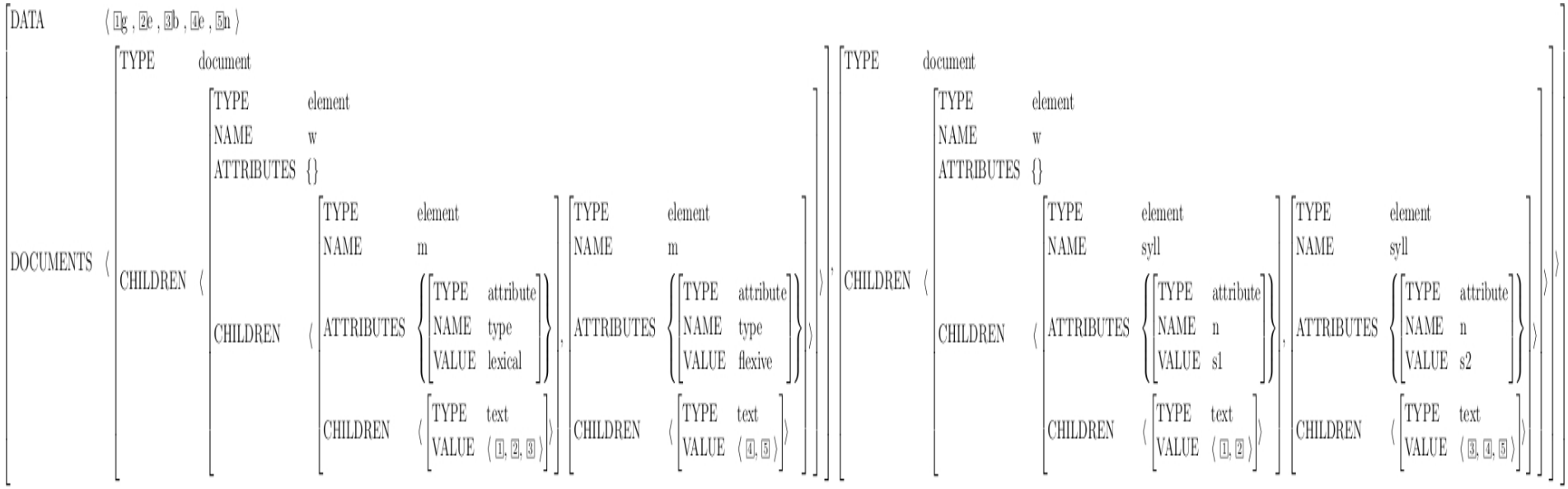
- top-level features are DATA (optional) and DOCUMENTS; both take collection values of the list kind and use a flat notation variant that makes use of "notational sugar" as provided by the TEI Guidelines
- the latter has a list of feature structures that represent the content of the respective documents to be represented

Representation alternative 2 as TEI FS

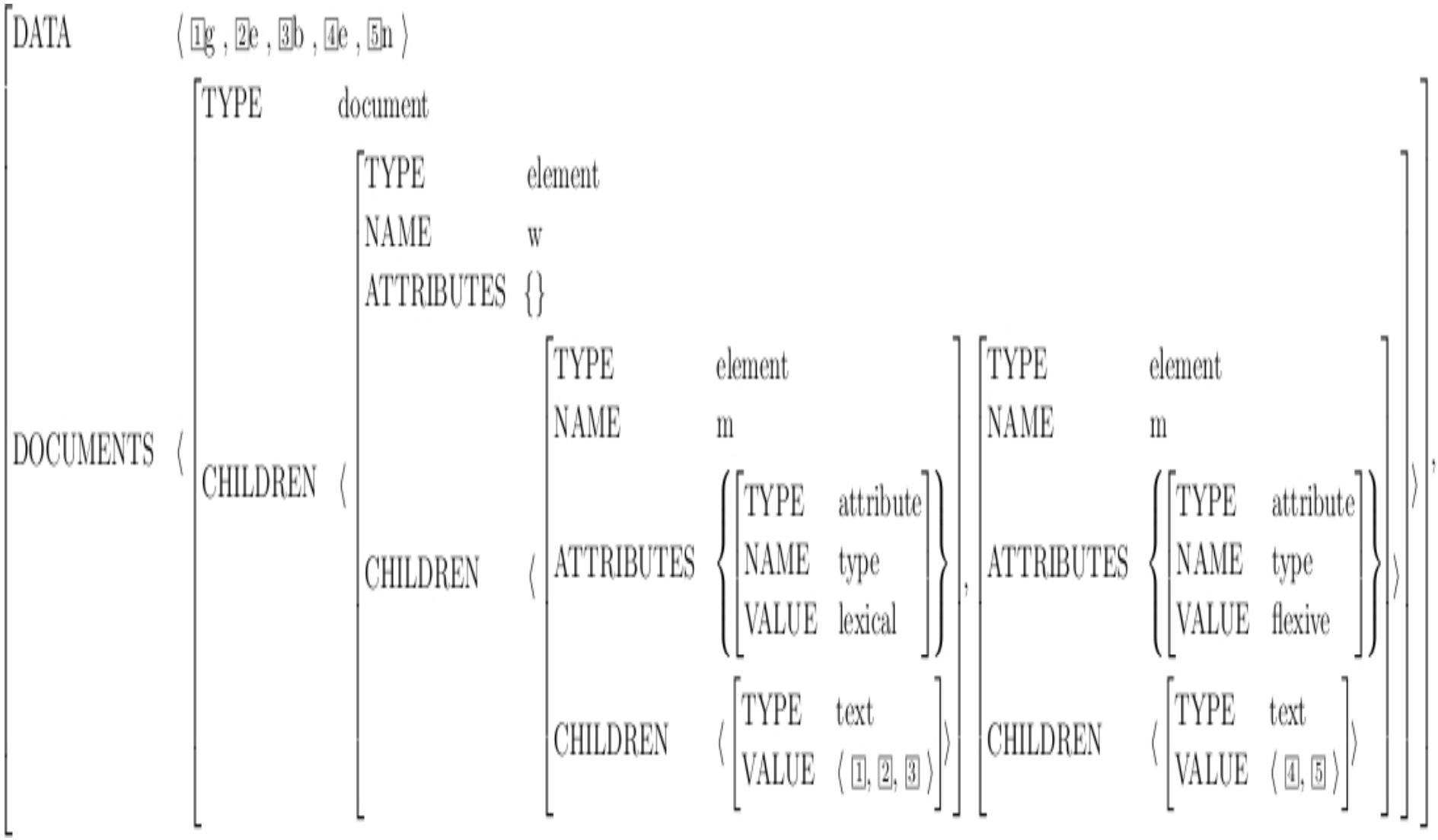
Mitglied der Leibniz Gemeinschaft

```
<fs>
 <f name="DATA">
  <vColl org="list">
   <vLabel name="1"><string>g</string></vLabel>
   <vLabel name="2"><string>e</string></vLabel>
   <vLabel name="3"><string>b</string></vLabel>
   <vLabel name="4"><string>e</string></vLabel>
   <vLabel name="5"><string>n</string></vLabel>
  </vColl> </f>
 <f name="DOCUMENTS">
  <vColl org="list"><fs> ... </fs> <fs> ... </fs></vColl>
    </f></fs>
```

Second representation alternative: AVM notation

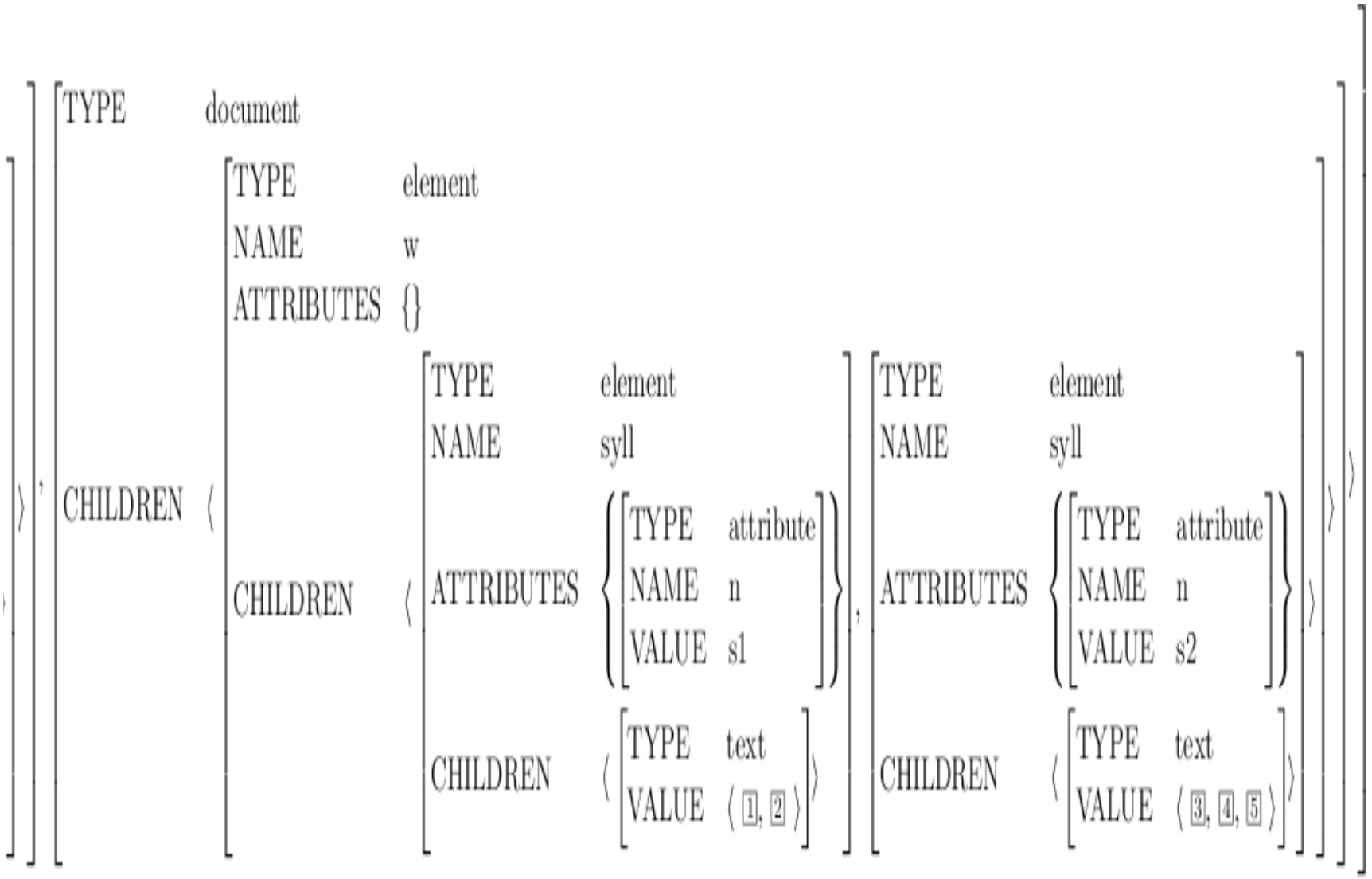


zoom1


DATA ⟨ 1g , 2e , 3b , 4e , 5n ⟩
DOCUMENTS ⟨
TYPE document
CHILDREN ⟨
TYPE element
NAME w
ATTRIBUTES {}
CHILDREN ⟨
TYPE element
NAME m
ATTRIBUTES { TYPE attribute, NAME type, VALUE lexical }
CHILDREN ⟨ TYPE text, VALUE ⟨ 1, 2, 3 ⟩ ⟩
,
TYPE element
NAME m
ATTRIBUTES { TYPE attribute, NAME type, VALUE flexive }
CHILDREN ⟨ TYPE text, VALUE ⟨ 4, 5 ⟩ ⟩
⟩ ⟩ ;

Zoom 2

Mitglied der Leibniz Gemeinschaft



Motivations for representation altern. 2

Mitglied der Leibniz Gemeinschaft

- perspective based on XDM, the XML data model for XPath2.0
  - node kinds distinguished: document, element, attribute, namespace, commentary, processing instruction, text;
  - node kind indicated by the TYPE feature for every node,
  - hierarchical relations indicated by CHILDREN feature on element- and document nodes,
  - other features used as appr.: NAME, ATTRIBUTES, VALUE
- systematic: every FS embedded below DOCUMENTS represents a certain node in the XDM tree model view of XML document instances

Aspects of the XSLT Implementation

Mitglied der Leibniz Gemeinschaft

- the program was written for the multiple annotation documents scenario, but it works with generic XML documents and single document instances, as well
- it is based on XSLT 2.0
- stylesheet parameters (defaults exist) in order to fine-tune aspects of the processing and details of the result format

Indexing

Mitglied der Leibniz Gemeinschaft

- the transformation of multiply annotated documents is based on indexing: making the implicit linkages among the documents explicit using structure sharing
- idea: indexing can be implemented by determining the relative position of the respective textual character among the other characters in the document = bound to be constant across all multiple annotation documents
- but: this is rather costly; we run into problems when processing large documents

4. Summary and Outlook

Up to this point …

- introduction to the format and domains of application
- informal introduction to FSs and their notation as proposed by the TEI (plus AVMs and graphs)
- representation alternatives wrt. multiply annotated documents as TEI FSs; generic XML documents are represented in the same way, just without the indexing
- characterization of the XSLT stylesheet for the mapping

Now that we can use FSs for XML documents …

- can we use tools provided by computational linguists?
- can operations like unification and generalization be applied to accordingly represented corpora or documents?

Unification

- two scenarios: multiple annotation and generic XML
- "direct" unification of respective encoded documents:
  - values under DATA unify for multiply annotated data, but usually not for arbitrary XML documents
  - values under DOCUMENTS usually do not unify for documents in both settings
- but: unification can be put to use in the context of rules that build bigger structures from smaller ones (or vice versa) and many linguistic applications make such use of unification
- we are free to write rules that make use of unification just the way we need it: this is very flexible

Generalization

- can be applied to show what is common to document representations
- the result will not be a complete representation of the information of both documents, but the information they have in common

Where are we now?

- One of the anonymous reviewers stated that (s)he thinks that the paper's "strength is as a thought experiment that has not provided quite the breakthrough that was hoped for it; yet interesting things have been learned and observed."
- we do agree totally: there are several drawbacks, but it seems that there are at least some open questions that remain to be investigated

Further possibilities

Mitglied der Leibniz Gemeinschaft

- we are free to write rules that make use of unification just the way we need it -> very flexible!
- generalization: can be applied to show what is common to document representations (not a complete representation of the information of both documents, of course)
- most promising perspective:
  Since feature structures are used when bare text is processed, the representation on annotated documents with FS has the potential to bridge the gap between the information given by annotations and the information contained in the text content

TEI Feature Structures as a Representation Format for Multiple Annotation and Generic XML Documents

Jens Stegmann

Bielefeld University

Andreas Witt

Institute for the German Language, Mannheim