ATOLL - INRIA Rocquencourt

http://atoll.inria.fr

Structures de Traits
Présentation et Commentaires

Éric de la Clergerie

Eric.De_La_Clergerie@inria.fr

Réunion RNIL

AFNOR – 23 Septembre 2003

Contexte

- Les structures de traits [Feature Structure] sont un méchanisme très générique (neutre) en Linguistique pour décrire des paires (propriété,valeurs).
- Déjà un chapitre sur la Représentation de FS (FSR) dans TEI, puis utilisation dans TAGML
- Proposition de normalisation dans ISOTC37SC4, conduite par Kiyong Lee (Corée)
- Committee Draft (CD) accepté sur FSR mais attente de commentaires supplémentaires.
  Rédaction d'un premier jet de commentaires ; préparation d'une réponse française
- Attente d'un chapitre sur la Déclaration de FS (FSD)
  Quelques propositions
- Effort de convergence avec TEI

Plusieurs vues des FS

- Vue matricielle AVM, la plus courante : Morpho-Syntaxe, HPSG, ...
- Vue Graphe « Acyclique » Dirigé [DAG]
- Vue Chemin, en LFG
  non couvert dans le Draft actuel, propositions dans mes commentaires

Un exemple AVM simple

Une simple liste de propriétés / valeurs atomiques,
éventuellement avec disjunction de valeurs atomiques

$$\begin{bmatrix} \text{number} & \text{sing} \\ \text{pers} & 3 \\ \text{mode} & \text{ind} \\ \text{tense} & \text{present} \\ \text{transitive} & + \\ \text{passive} & - \end{bmatrix}$$

Un exemple AVM complexe (Lexique HPSG)

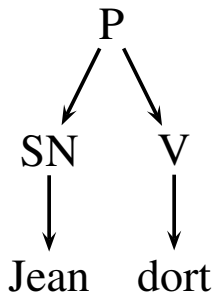
$$\begin{bmatrix} \text{PHON} & \langle \text{rouge} \rangle \\ \text{SYNSEM} & \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{TETE} & \begin{bmatrix} \text{MAJ} & \text{Adj} \\ \text{PRD} & - \\ \text{MOD} & \begin{bmatrix} \text{N'} : \begin{bmatrix} \text{INDEX} & \boxed{1} \\ \text{RESTR} & \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}_{\text{adj}} \\ \text{VALENCE} & \langle\rangle \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{INDEX} & \boxed{1}\begin{bmatrix} \text{NOMB} & \text{sing} \end{bmatrix} \\ \text{RESTR} & \left\{ \begin{bmatrix} \text{QUANTS} & \langle\rangle \\ \text{NUCLEUS} & \begin{bmatrix} \text{ARG} & \boxed{1} \end{bmatrix}_{\text{rouge}} \end{bmatrix} \right\} \cup \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}_{\text{synsem}} \end{bmatrix}$$

Unification de FS

$$\begin{bmatrix} \text{Det} \begin{bmatrix} \text{Accord } \boxed{1} \begin{bmatrix} \text{Num sing} \end{bmatrix} \end{bmatrix} \\ \text{Nom} \begin{bmatrix} \text{Accord } \boxed{1} \\ \text{Cat N} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \text{Det} \begin{bmatrix} \text{Accord} \begin{bmatrix} \text{Genre Masc} \end{bmatrix} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \text{Det} \begin{bmatrix} \text{Accord } \boxed{1} \begin{bmatrix} \text{Num sing} \\ \text{Genre masc} \end{bmatrix} \end{bmatrix} \\ \text{Nom} \begin{bmatrix} \text{Accord } \boxed{1} \\ \text{Cat N} \end{bmatrix} \end{bmatrix}$$

Vue par chemin (Exemple LFG)

P $\longrightarrow$	SN	V	SN $\longrightarrow$	Jean	V $\longrightarrow$	dort	
	(↑Suj)=↓	↑=↓		(↑Num)= sing		(↑Suj Num)= sing	
				(↑Genre)= masc		(↑Suj Pers)= 3	
				(↑Pred)= 'Jean'		(↑Mode)= indicatif	
						(↑Pred)= 'dormir<Suj>'	



$$S\begin{bmatrix} \text{uj}\begin{bmatrix}\text{red 'Jean'}\\ \text{Genre masc}\\ \text{Num sing}\end{bmatrix}_P \\ \text{Pred 'dormir}\langle\text{Suj}\rangle\text{'} \\ \text{Mode indicatif}\end{bmatrix}$$

Comment aborder les FS

Introduire plusieurs niveaux de complexité dans la norme pour en faciliter l'appropriation.

1. FS non typées à valeurs atomiques (fs, f, sym)

2. Disjunctions de valeurs atomiques (vAlt)

3. Bibliothèques de valeurs et referencements compacts (fvLib, fLib, fsLib)

4. Typage des FS

5. Récursion (fs dans fs) et Réentrance (partage, variables)

6. Valeurs complexes (alternative générales, bags, lists, sets)

7. Déclaration des FS

Format simple

number	sing
pers	3
mode	ind
tense	present
transitive	+
passive	-

```
<fs>
  <f name="number"><sym value="sing"/></f>
  <f name="pers"><sym value="3"/></f>
  <f name="mode"><sym value="ind"/></f>
  <f name="tense"><sym value="present"/></f>
  <f name="transitive"><plus/></f>
  <f name="passive"><minus/></f>
</fs>
```

Valeurs atomiques

Le Draft (et la TEI) introduit toutes sortes de valeurs atomiques :

sym, plus, minus, nbr, msr, rate, str

- Effet bestiaire
- Pas de garantie d'avoir une liste close (pourquoi pas date)
- Comment échapper vers ses propres valeurs atomiques ?
- S'appuyer sur XML schéma pour instancier ses propres types atomiques

Alternatives simples

Proposition d'une balise spécifique pour les disjunctions de valeurs (atomique ?)

```
<fs>
  <f name="number"><sym value="sing"/></f>
  <f name="pers"><sym value="3"/></f>
  <f name="mode">
      <vAlt>
          <sym value="ind"/>
          <sym value="subj"/>
      </vAlt>
  </f>
  <f name="tense"><sym value="present"/></f>
  <f name="transitive"><plus/></f>
  <f name="passive"><minus/></f>
</fs>
```

Note : Une certaine redondance avec la disjonction générique alt

Bibliothèques

- Possibilité de nommer valeur, paires traits-valeurs et FS ;
- regroupement en bibliothèque ;
- modes compacts de référencement

```
<fvLib> <!-- Value library -->
    <sym id="n" value="n"/>
    <sym id="sing" value="sing"/>
    <sym id="fem" value="fem"/>
</fvLib>
<fLib> <!-- Feature-Value library -->
  <f id="pos@n" name="pos" fVal="n"/>
  <f id="num@sing" name="num" fVal="sing"/>
  <f id="gen@fem" name="gen" fVal="fem"/>
</fLib>
```

```
<w entry="pomme␣de␣terre" tokens="1␣2␣3"> <!-- potato -->
     <fs feats="pos@n␣num@sing␣gen@fem"/>
</w>
```

Typage

Pas de difficulté sur le typage, si pas d'aspect déclaration de FS (FSD).

```
<w entry="pomme de terre" tokens="1 2 3"> <!-- potato -->
    <fs type="noun" feats="num@sing gen@fem" />
</w>
```

Recursion – Enchâssement

Pas de difficultés majeures : fs comme valeur de f

```
<fs>
  <f name="phon"><sym value="rouge"/></f>
  <f name="synsem">
      <fs type="synsem">
          <f name="loc">
              <fs type="loc">
                  ...
              </fs>
          </f>
      </fs>
  </f>
</fs>
```

Reentrance, partage, variables

Le point difficile !

$$\begin{bmatrix} \text{Det} & \begin{bmatrix} \text{Accord} & \boxed{1} \begin{bmatrix} \text{Num} & \text{sing} \end{bmatrix} \\ \text{Cat} & \text{Det} \end{bmatrix} \\ \text{Nom} & \begin{bmatrix} \text{Accord} & \boxed{1} \\ \text{Cat} & \text{N} \end{bmatrix} \end{bmatrix}$$

- Relier deux points ou plus d'une FS (voire de FS différentes)
- On relie les champs (f) ou les valeurs ?

```
<f name="accord" var="@1"><fs>...</fs></f>
...
<f name="accord" var="@1"/>
```

```
<f name="accord"><var name="@1"><fs>...</fs></var></f>
...
<f name="accord"><var name="@1"/></f>
```

- Deuxième alternative permet des variables dans disjunctions (et valeurs complexes)

```
<f name="subcat" org="list">
    <var name="@Obj"><fs type="noun">..</fs></var>
    <var name="@PPObj"><fs type="pp">..</fs></var>
</f>
```

- L'utilisation de Id unique gène l'utilisation des bibliothèques : souhaite réutilisation de FS de bibliothèques avec renommage des points partagées.
- Les notations par référence XPath sont une solution mais cassent la symétrie du partage et sont lourdes

```
<fs>
 <f name="det">
      <fs><f name="accord"><fs>...</fs></f></fs>
 </f>
 <f name="nom">
      <fs><f name="accord" share="../../f[@name=det]/fs/f"/></fs>
 </f>
</fs>
```

Reentrance : quelles alternatives ?

- Alternative 1 : Utiliser des identifiants non uniques et gérer leur portée par l'application.

- Alternative 2 : Specifier le scope des variables
  Existence de tels méchanismes dans XML ?

```
<fs fsvars="@1">
  <f name="det">
      <fs><f name="accord"><var name="@1"><fs>...</fs></var></f></fs>
  </f>
  <f name="nom">
      <fs><f name="accord"><var name="@1"/></f></fs>
  </f>
</fs>
```

- Alternative 3 : Notation chemin en XPath – Symétrie et grande flexibilité

```
<fs>
  <f name="det">
       <fs><f name="acc"><fs>..</fs></f></fs>
  </f>
  <f name="nom">
       <fs><f name="acc"/></fs>
  </f>
  <fseq share="f[name=det]/fs/f[name=acc] f[name=nom]/fs/f[name=acc]"/>
</fs>
```

Constat : Pas réellement de solution simple et élégante.

Valeurs complexes

Groupes de valeurs organisés en set, bag et list

Pas de difficultés majeures, sauf en relation avec la réentrance.

```
<f name="subcat" org="set">
    <var name="@Obj"><fs type="noun">..</fs></var>
    <var name="@PPObj"><fs type="pp">..</fs></var>
</f>
```

Note : Le Draft ne dit rien sur la concaténation de valeurs complexes (listes, sets, bags.

$$\left[\text{RESTR} \left\{\left[\begin{array}{l}\text{QUANTS} \quad \langle\rangle \\ \text{NUCLEUS} \left[\begin{array}{l}\text{ARG} \quad \boxed{1} \\ \text{rouge}\end{array}\right]\end{array}\right]\right\} \cup \boxed{2}\right]$$

Notation pour les chemins

- Adéquat pour de nombreux formalismes, dont LFG
- Permet d'ajouter des infos de partages en dehors de la structure (incrémentalité)
- Notation compact possible

```
<path><f name="subj"><f name="num"/></f></path>
<path name="subj num"/>
```

- Pourrait même être utilisé comme f, voire fusion de f en path

```
<fs>
  <path name="subj num"><sym value="sing"/></path>
  <f name="mode"><sym value="ind"/></f>
</fs>
```

```
<fs>
  <path name="subj num"><sym value="sing"/></path>
  <path name="mode"><sym value="ind"/></path>
</fs>
```

Notation chemin

- Permet éventuellement une plus grande compacité pour la réentrance

```
<fseq share="f[name=det]/fs/f[name=acc] f[name=nom]/fs/f[name=acc]" />
```

```
<fseq path1="det acc" path2="nom acc" />
```

- Extension envisageable à une algébre des chemins (disjonction et kleene)
  par exemple pour l'incertitude fonctionnelle LFG

```
<path> <!-- k(g|h)* -->
    <f name="k">
        <fstar>
            <alt>
                <f name="g" />
                <f name="h" />
            </alt>
        </fstar>
    </f>
</path>
```

Declaration de Structures de Traits [FSD]

- Revient à déclarer les structures admissibles les plus générales possibles
- En première approximation, proche des méchanismes de bibliothèques :
  bibliothèques des ensembles de valeurs et de structures les plus générales possibles.

```
<fvLib>
  <vAlt id="gender_range"><sym value="masc"/><sym value="fem"/></vAlt>
  <vAlt id="number_range"><sym value="sing"/><sym value="pl"/></vAlt>
</fvLib>
<fsLib>
  <fs id="maxagr" type="agr">
    <f name="gender" fVal="gender_range">
    <f name="number" fVal="number_range">
  </fs>
</fsLib>
```

Hierarchies de types à la Carpenter

Utilisées dans ALE pour HPSG. Peuvent servir de base pour FSD.

```
bot sub [string,list,cat,synsem].
    string escape symbol.
    cat sub [s,np,vp,det,n].
        s sub []. np sub []. vp sub [].
        det sub []. n sub [].
    synsem sub [phrase,lexeme] intro [cat:cat].
        phrase sub [root] intro [args:list].
            root sub [] intro [cat:s].
        lexeme sub [] intro [orth:string].
    list sub [ne_list,e_list].
        ne_list sub [] intro [hd:bot,tl:list].
        e_list sub [].
```

- Un type peut introduire des sous-types
- Un type peut introduire ou préciser un trait et le type de sa valeur
- héritage multiple possible
- les types introduisant un trait f ont un type plus général

Déclaration de structure de traits

```
<FSHierarchy type="list_of_dates">
  <FStype name="top" id="top">
    <FStype name="list" id="list">
        <FStype name="e_list"/>
        <FStype name="ne_list">
           <f name="hd"> <FSatomic type="date"/> </f>
           <f name="tl"> <FStype idref="list"/> </f>
        </FStype>
    </FStype>
  </FStype>
</FSHierarchy>
```

```
<FSHierarchy type="family">
  <FStype name="top" id="top">
    <FStype name="person" id="person">
        <f name="first_name"> <FSatomic type="string"/> </f>
        <f name="last_name"> <FStype idref="string"/> </f>
        <FStype name="married">
          <fs type="married" var="X">
              <f name="last_name" var="N"/>
              <f name="spouse">
                  <fs type="married">
                      <f name="last_name" var="N"/>
                      <f name="spouse" var="X"/>
                  </fs>
              </f>
          </fs>
        </FStype>
    </FStype>
  </FStype>
</FSHierarchy>
```

Combiner hiérarchie et formes maximales

```
<FSHierarchy type="family">
  <FStype name="top" id="top">
    <FStype name="person" id="person">
        <f name="first_name"> <FSatomic type="string"/> </f>
        <f name="last_name"> <FStype idref="string"/> </f>
        <FStype name="married">
          <fs type="married" var="X">
              <f name="last_name" var="N"/>
              <f name="spouse">
                  <fs type="married">
                      <f name="last_name" var="N"/>
                      <f name="spouse" var="X"/>
                  </fs>
              </f>
          </fs>
        </FStype>
    </FStype>
  </FStype>
</FSHierarchy>
```

Un peu de « philosophie »

- FS neutre mais très puissant
- Reflète en partie XML : FSD $\sim$ DTD ou Schéma, Réentrance $\sim$ XPath, FS $\sim$ XML.
- $\Rightarrow$ faire attention à ne pas trop mettre dans FS, qui peut être géré par les méchanismes XML
- Éventuellement, identifier limites XML : partage sous scope
  portée des identifiants dans XML