Problems in "symbol-based" grammars

Det Noun Verb Noun Det
Adj
J JJ J

Lexicon:
the: Det (definite)
little: Adj (size)
orange: Adj (colour)
Noun (fruit)
ducks: Noun (animal)
swallow: Verb (action)
flies: Noun (animal)
Verb (action)
Verb (action)

Agreement

• http://www.w3.org/TR/speech-grammar/#S6.2 (aug. 01)
  “The morphological variations of a token may depend upon the grammatical class of the
token (e.g. verb, noun, article, adjective). Morphological rules are intrinsically
language-specific and some languages have much richer morphological behavior than
others.” [...]. Example (pseudo format)
< morph > dog </ morph > < morph > drink </ morph > faster
dog:: dog | dogs
drink:: drink | drank | drunk | drinks

• Status: The Working Group is not aware of any existing grammar format that
supports this kind of morphological inference by a speech recognizer for a grammar.

• The grammar format "supporting" this "reference" problem is unification
  grammar!!!


Actual origin in unpublished documents by M. Kay in the seventies

Classical presentations:

UNIFICATION GRAMMAR


Actual origin in unpublished documents by M. Kay in the seventies

Classical presentations:

UNIFICATION GRAMMAR

PSG/BNF
• symbols
• parsing: symbols are string-compared
• comparison of two symbols returns TRUE or FALSE
• the set of symbols is static

APSG
• feature sets
• parsing: feature sets are unified
• unification of two feature sets returns a new feature set or NULL
• the set of feature sets is dynamic
**Terminology:**
- **attribute:** lex, cat
- **value:** s, np, vp, fly ...
- **feature set (compound feature):** \{lex=fly, cat=n\} ...
- **atomic feature:** lex=fly
- **complex feature:** agreement={prs=third, nb=plur}
- **constant feature:** prs=third
- **variable feature:** prs=$A

* NOT supported by all unification grammars (e.g. by the CPK NLP Suite)

**Unification:** the basic operation on compound features

\[
\begin{align*}
\{\text{cat}=n\} \cup \{\text{lex}=\text{flies}, \text{cat}=n\} &= \{\text{lex}=\text{flies}, \text{cat}=n\} \\
\{\text{cat}=n\} \cup \{\text{lex}=\text{flies}, \text{cat}=n\} &= \text{NULL} \\
\{\text{cat}=\text{pron}, \text{nb}=\text{plur}\} \cup \{\text{lex}=\text{you}, \text{cat}=\text{pron}\} &= \{\text{lex}=\text{you}, \text{cat}=\text{pron}, \text{nb}=\text{plur}\} \\
\{\text{cat}=\text{pron}, \text{nb}=\text{sing}\} \cup \{\text{lex}=\text{you}, \text{cat}=\text{pron}\} &= \{\text{lex}=\text{you}, \text{cat}=\text{pron}, \text{nb}=\text{sing}\} \\
\{\text{cat}=\text{pron}, \text{nb}=\text{plur}\} \cup \{\text{lex}=1, \text{cat}=\text{pron}, \text{nb}=\text{sing}\} &= \text{NULL}
\end{align*}
\]

**Unification and variables 1):**

\[
\begin{align*}
&\{\text{cat}=\text{pron}, \text{nb}=$N\}\cup \{\text{lex}=\text{I}, \text{cat}=\text{pron}, \text{nb}=\text{sing}\} = \{\text{lex}=\text{I}, \text{cat}=\text{pron}, \text{nb}=\text{sing}\} \\
&\{\text{cat}=\text{pron}, \text{nb}=$N\}\cup \{\text{lex}=\text{you}, \text{cat}=\text{pron}\} = \{\text{lex}=\text{you}, \text{cat}=\text{pron}, \text{nb}=$N\}
\end{align*}
\]

**Unification and variables 2):**

\[
\begin{align*}
&\{\text{cat}=\text{np}, \text{nb}=$N\}\cup \{\text{cat}=\text{vp}, \text{nb}=$N\} \\
&\{\text{cat}=\text{np}, \text{nb}=\text{sing}\}\cup \{\text{lex}=\text{I}, \text{cat}=\text{pron}, \text{nb}=\text{sing}\} = \{\text{cat}=\text{np}, \text{nb}=\text{sing}\}\cup \{\text{cat}=\text{vp}, \text{nb}=\text{sing}\}\cup \{\text{lex}=\text{I}, \text{cat}=\text{pron}, \text{nb}=\text{sing}\}\cup \{\text{lex}=\text{np}, \text{cat}=\text{np}, \text{nb}=$N\}\cup \{\text{lex}=\text{vp}, \text{cat}=\text{vp}, \text{nb}=$N\}
\end{align*}
\]

**Unification and variables 3):**

\[
\begin{align*}
&\{\text{cat}=\text{s}\}\cup \{\text{cat}=\text{np}, \text{nb}=$N\}, \{\text{cat}=\text{vp}, \text{nb}=$N\}\} \\
&\{\text{cat}=\text{np}, \text{nb}=\text{sing}\}\cup \{\text{cat}=\text{np}, \text{nb}=\text{sing}\}\cup \{\text{cat}=\text{vp}, \text{nb}=\text{sing}\}\cup \{\text{cat}=\text{vp}, \text{nb}=\text{sing}\}\cup \{\text{cat}=\text{s}\}
\end{align*}
\]

**Unification and variables 4):**

Agreement: a variable occurs twice or more in the body of a rule:

\[
\begin{align*}
&\{\text{cat}=$s$, \text{stype}=\text{decl}\} \\
&\{\text{cat}=\text{np}, \text{prs}=$P$, \text{nb}=$N\}, \{\text{cat}=\text{np}, \text{prs}=$P$, \text{nb}=$N\}\} \\
&\{\text{cat}=\text{np}, \text{prs}=$P$, \text{nb}=$N\}, \{\text{cat}=\text{np}, \text{prs}=$P$, \text{nb}=$N\}\}
\end{align*}
\]

i.e.: subject - predicate agreement in person and number:

I am, you are, the man is, the men are, *I is, *you am...
Unification and variables 5:

Percolation: a variable occurs once in the head and once (or more) in the body of a rule:

\[
\begin{align*}
{\text{cat}=np, \text{prs}=\text{third}, \text{nb}=\$N,} \\
^{\text{cat}=\text{det}}, & \quad /* \text{optional determiner } \text{a}, \text{the} */ \\
{\text{cat}=\text{adj}}, & \quad /* \text{zero or more adjectives } */ \\
{\text{cat}=\text{n}, \text{nb}=\$N}, & \quad /* \text{obligatory noun } */
\end{align*}
\]

i.e.: the number of an NP (subject, object) is the number of the noun contained in the NP: the man (sing), the large tall man (sing), men (plur.), the men (plur.).

The special “cat”-feature, obligatory in most formalisms:

- **Historical reasons:** In the earliest approaches, features were introduced as additional “feature constraints” to the PSG:
- **Parsin:** many efficient parsing algorithms developed for normal context-free grammars (like Earley) can only be applied on feature-based grammars with at least one obligatory feature.
- **Prevents unification from ‘exploding’**

Another special feature: “lex”

- In some formalisms, lexical rules consist of identifier+feature set: “man”: \{cat=noun, number=sing, case=non-genitive\}
- In other formalisms, the identifier is a feature, bound to the attribute “lex”, that can participate in unification the normal way: \{lex=man, cat=noun, number=sing, case=non-genitive\}

Shieber 1986:

- **General demands on grammar formalisms**
  - Linguistic felicity
  - Expressiveness
  - Computational effectiveness
- **Characteristics of Unification Grammars**
  - Surface-based
  - Informational
  - Inductive
  - Declarative

---

**CPK NLP Suite**

<table>
<thead>
<tr>
<th>cpk</th>
<th>nlp</th>
<th>suite</th>
</tr>
</thead>
<tbody>
<tr>
<td>aps</td>
<td>psg</td>
<td>icm</td>
</tr>
<tr>
<td>psg</td>
<td>apspars</td>
<td>psgpars</td>
</tr>
<tr>
<td>trec</td>
<td>apstrec</td>
<td></td>
</tr>
<tr>
<td>tslu</td>
<td>apstslu</td>
<td></td>
</tr>
<tr>
<td>conv</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**VoiceXml 1.0**
W3C Voice Browser Activity
Status Feb 2002 (http://www.w3.org/Voice/)

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech recognition grammars</td>
<td>Speech Recognition Grammar Markup Language</td>
</tr>
<tr>
<td>Voice dialogs</td>
<td>VoiceXML 2.0, STT, &quot;Momordinae of understanding&quot;</td>
</tr>
<tr>
<td>Speech synthesis</td>
<td>Speech Synthesis Markup Language</td>
</tr>
<tr>
<td>Natural language representation</td>
<td>Natural Language Semantics Markup Language</td>
</tr>
<tr>
<td>Multimodal systems</td>
<td>NEW: given up</td>
</tr>
<tr>
<td>Reusable dialog components</td>
<td>NEW: given up</td>
</tr>
</tbody>
</table>

What’s new?
- Real standard
- Official support for
  - Speech Recognition Grammar Markup Language
  - Speech Synthesis Markup Language
- Planned integration with other W3C Voice Browser activities (WAP, multimodal applications...)

HTML “Hello World”
Source
```html
<head> [...]</head>
<body>
This is just plain HTML:
<br><u>underlined text</u>
<br><i>italic</i>
<br>And here is a picture:
<img SRC="tb.jpg">
</body>
```
Interpreted (e.g. Netscape)

VoiceXML “Hello World”
Source
```xml
<menu>
<prompt>
Say one of: <enumerate/>
</prompt>
<choice next="http://www.sports.example/start.vxml">
Sports
</choice>
<choice next="http://www.weather.example/intro.vxml">
Weather
</choice>
<noinput>Please say one of <enumerate/></noinput>
</menu>
```
Interpreted

Interpreters & Tutorials
- Interpreters
  - IBM WebSphere Voice Server SDK
  - Others
    - see http://www.w3.org/Voice/
- Tutorials
  - http://www.w3.org/Voice/Guide/
VoiceXML has good dialog components

- Menu dialog
  - Used when a single value/turn leads to the next action
  - Good abstraction for menus and list selection
- Form dialog
  - Used when multiple fields must be filled before the next action
  - Good abstraction for information gathering for a transaction

Dialogue Control: Menu

```xml
<menu>
  <prompt>
    Say one of: <enumerate/>
  </prompt>
  <choice next="http://www.sports.example/start.vxml">
    Sports
  </choice>
  <choice next="http://www.weather.example/intro.vxml">
    Weather
  </choice>
  <noinput>Please say one of <enumerate/></noinput>
</menu>
```

Dialogue Control: Form

```xml
<form>
  <field name="city">
    <prompt>Where do you want to travel to?</prompt>
    <option>Edinburgh</option>
    <option>New York</option>
  </field>
  <field name="travellers" type="number">
    <prompt>How many are travelling to <value expr="city'/>?</prompt>
  </field>
  <block>
    <submit next="http://localhost/handler" namelist="city travellers"/>
  </block>
</form>
```

VoiceXML integrates with the Internet

- Designed for a client/server environment
- Transactions are handled by an application server using existing web protocols (HTTP, JSP, CGI, ASP, JavaServlets, …)
- VoiceXML documents, grammars, and sound files are referenced by URLs
- Will probably appeal to web programmers

VoiceXML makes highly dynamic dialogues possible

- VoiceXML documents can be generated “on the fly” by the server
- Form interpretation allows multiple paths through the dialogue
- Embedded ECMAScript allows “wild” constructions
  - expression can calculate the next state
  - field variables can be changed

Apparent simplicity hides the complexity of dialogue design

- Making a spoken dialogue application is still a complex task
- Speech recognition performance is still a limiting factor
- A deeper understanding of the underlying technologies is still required
VoiceXML applications may have limited portability

- Differences in the implementation of VoiceXML portals is likely to impact the performance of VoiceXML applications.
- Speech recognition performance will vary between portals.
- Differences in speech synthesis will influence prompting, which may in turn influence user responses.

Verification of dialogue applications can be difficult

- The form filling may lead to multiple dialogue paths.
- The use of embedded ECMAScript in a dialogue increases the risk of bugs surfacing as runtime errors.
- Dialogues may execute remotely in an environment which is outside the control of the dialogue programmer.

VoiceXML is not suitable for embedded dialogues

- Designed for a client/server environment.
- Needs a large support structure (a browser).
  - Generic speech recognition: Dictation mode, JSGF, word spotting.
  - Text to Speech Synthesis.
- Verification is difficult/costly.
  - Form filling leads to multiple dialogue paths.
  - ECMAScript interpretation.

VoiceXML summary

- Sets a standard for voice portals - supported by W3C (new!)
- We should not expect it to improve the performance of spoken dialogue applications.
- We may see a proliferation of poorly performing spoken dialogue applications.