Conceptual Logical Structures:
The Lexico-Conceptual Linkage in Role and Reference Grammar

Carlos Periñán¹ & Ricardo Mairal²
¹ Universidad Católica San Antonio (Spain)
² UNED (Spain)
Logical Structures

Betty asked Bill for an apple.

\[
\text{IF DECL} \xleftarrow{\text{TNS PAST}} \left[\text{do'} (\text{Betty, say'} (\text{Betty, Bill}))\right] \text{PURP} \left[\text{do'} (\text{Bill, 0})\right]
\]

\text{CAUSE} [\text{BECOME have'} (\text{Betty, apple})]>>>

Logical Structures

Arreglar

(i) put into a proper or systematic order: (= ARRANGE)

  e.g. Mi madre arregló las flores del jarrón

  [do’ (x, 0)] CAUSE [BECOME arranged’ (y)]

(ii) restore by replacing a part or putting together what is torn or broken (= REPAIR)

  e.g. Mi padre arregló el televisor

  [do’ (x, 0)] CAUSE [BECOME repaired’ (y)]
Conceptual Logical Structure

Betty asked Bill for an apple.

\[
\langle \text{IF DECL} \rangle_{\text{TNS PAST}} \langle \text{do (\%BETTY\_00 Theme, [+REQUEST\_01 (\%BETTY\_00 Theme, \%BILL\_00 Goal)])}] \rangle_{\text{PURP}} \langle \text{do (\%BILL\_00 Goal, 0)} \rangle_{\text{CAUSE}} \langle \text{BECOME +REQUEST\_01 (\%BETTY\_00 Theme, +APPLE\_00 Referent)} \rangle_{\text{>>>}}
\]

- Language-independent representation
- Lexico-conceptual linkage
FunGramKB Lexicon\textsubscript{English}

- dry

FunGramKB Lexicon\textsubscript{Spanish}

- secar
LCM CORE GRAMMAR:

AktionsArt:

☐ State  
☐ Activity  
☑ Accomplishment  
☐ Achievement

You determine the canonical lexical class(es) of the verb.

Variables: $x, y$

Idiosyncratic features:

$[MR \leftarrow \text{no value selected} \rightarrow ], [U = \leftarrow \text{no value selected} \rightarrow ]$

Thematic frame mapping:

$x = \text{Theme}, y = \text{Referent}, z = \text{[no function]}$

Lexical Template:

A REMINDER OF FUNGRAMKB PARTICIPANTS:

THEME: Entity that transforms another entity.
REFERENT: Entity that is transformed by another entity.
LCM CORE GRAMMAR:

AktionsArt:

- State
- Activity
- Accomplishment
- Achievement

You determine the canonical lexical class(es) of the verb.

Variables: x, y

Idiosyncratic features:

[MR <-- no value selected -->], [U <-- no value selected -->]

Thematic frame mapping:

x = Theme, y = Referent, z = [no function]

Lexical Template:

A REMINDER OF FUNGRAMKB PARTICIPANTS:

THEME: Entity that transforms another entity.

REFERENT: Entity that is transformed by another entity.
AktionsArt:

You determine the canonical lexical class(es) of the verb.

Variables: \(x, y\)

Idiosyncratic features:

\([\text{MR} \leftarrow \text{no value selected} \rightarrow], [\text{U} = \leftarrow \text{no value selected} \rightarrow]\)

Thematic frame mapping:

\(x = \text{Theme}, y = \text{Referent}, z = \text{[no function]}\)

Lexical Template:

**A REMINDER OF FUNGRAMKB PARTICIPANTS:**

**THEME:** Entity that transforms another entity.

**REFERENT:** Entity that is transformed by another entity.
**LCM Core Grammar:**

<table>
<thead>
<tr>
<th>AktionsArt:</th>
</tr>
</thead>
</table>

- State
- Activity
- Accomplishment
- Achievement

You determine the canonical lexical class(es) of the verb.

**Variables:**\n\[ x, y \]

**Idiosyncratic features:**
\[
[M_R \quad \text{[-- no value selected -->]}], [U = \quad \text{[-- no value selected -->}]
\]

**Thematic frame mapping:**
\[
X = \quad \text{Theme} \quad , \quad Y = \quad \text{Referent} \quad , \quad Z = \quad \text{[no function]}
\]

**Lexical Template:**

*A REMINDER OF FUNGRAMKB PARTICIPANTS:*

- **THEME:** Entity that transforms another entity.
- **REFERENT:** Entity that is transformed by another entity.
LCM CORE GRAMMAR:

AktionsArt:

- State
- Activity
- Accomplishment
- Achievement

You determine the canonical lexical class(es) of the verb.

Variables: \( x, y \)

Idiosyncratic features:

\[ [M_R] \quad [U] \]

Thematic frame mapping:

\( x = \text{Theme}, \quad y = \text{Referent}, \quad z = \text{[no function]} \)

Lexical Template:

A REMINDER OF FUNGRAMKB PARTICIPANTS:

THEME: Entity that transforms another entity.

REFERENT: Entity that is transformed by another entity.
Concept:
+OPEN_01

Thematic Frame:
(x1)Agent (x2: +DOOR_00 ^ +WINDOW_00)Theme (x3)Location (x4)Origin (x5)Goal

Thematic-Frame Mapping:
x = Agent, y = Theme
Aktionsart: Accomplishment
Causative Accomplishment

Variables: x, y
Idiosyncratic features: none
TF Mapping: x = Theme, y = Referent

BECOME +DRY_01 (yReferent)

[do (xTheme, 0) ] CAUSE [BECOME +DRY_01 (yReferent)]
<table>
<thead>
<tr>
<th></th>
<th>Symbol Representation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>&lt;C&gt; (&lt;VAR&gt;)</td>
<td>&lt;STA&gt;</td>
</tr>
<tr>
<td>Activity</td>
<td>do (x [&lt;STA&gt;])</td>
<td>&lt;ACT&gt;</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>BECOME &lt;STA&gt;</td>
<td>&lt;ACC&gt;</td>
</tr>
<tr>
<td>Achievement</td>
<td>INGR &lt;STA&gt;</td>
<td>&lt;ACH&gt;</td>
</tr>
<tr>
<td>Semelfactive</td>
<td>SEML &lt;STA&gt;</td>
<td>&lt;SEM&gt;</td>
</tr>
<tr>
<td>Active accomplishment</td>
<td>do (x [&lt;C&gt; (x)]) &amp; &lt;ACH&gt;</td>
<td>&lt;ACA&gt;</td>
</tr>
<tr>
<td>Causative state</td>
<td>[&lt;CLS&gt;] CAUSE [&lt;STA&gt;]</td>
<td>&lt;CSTA&gt;</td>
</tr>
<tr>
<td>Causative activity</td>
<td>[&lt;CLS&gt;] CAUSE [&lt;ACT&gt;]</td>
<td>&lt;CACT&gt;</td>
</tr>
<tr>
<td>Causative accomplishment</td>
<td>[&lt;CLS&gt;] CAUSE [&lt;ACC&gt;]</td>
<td>&lt;CACC&gt;</td>
</tr>
<tr>
<td>Causative achievement</td>
<td>[&lt;CLS&gt;] CAUSE [&lt;ACH&gt;]</td>
<td>&lt;CACH&gt;</td>
</tr>
<tr>
<td>Causative semelfactive</td>
<td>[&lt;CLS&gt;] CAUSE [&lt;SEM&gt;]</td>
<td>&lt;CSEM&gt;</td>
</tr>
<tr>
<td>Causative active accomplishment</td>
<td>[&lt;CLS&gt;] CAUSE [&lt;ACA&gt;]</td>
<td>&lt;CACA&gt;</td>
</tr>
</tbody>
</table>
CLS Constructor

1. Expanding LS metavariables
   (e.g. <STA>, <ACT>, <ACC>, <ACH>, <SEM>, <ACA>)

2. Inserting the concept
   (i.e. <C>)

3. Inserting variables
   (i.e. <VAR>)

4. Inserting thematic roles
[do' (x, 0) ] CAUSE [BECOME dry' (y)]
Aktionsart: Accomplishment
Causative Accomplishment
Variables: x, y
Idiosyncratic features: none
TF Mapping: x = Theme, y = Referent

BECOME +DRY_01 (yReferent)

[do (xTheme, 0) ] CAUSE [BECOME +DRY_01 (yReferent)]
Thematic Frame

(x1) Theme
(x2: +CORPUSCULAR_00) Referent

+DRY_01

Meaning Postulate

+(e1: +CHANGE_00 (x1) Theme
(x2) Referent (f1: (e2: n +BECOME_00
(x2) Theme (x3: +WET_00) Attribute)) Result)

FunGramKB Lexicon

FunGramKB Ontology

LCM Core Grammar

Aktionsart: Accomplishment
Causative Accomplishment
Variables: x, y
Idiosyncratic features: none
TF Mapping: x = Theme, y = Referent

BECOME +DRY_01 (y Referent)

[do (x Theme, 0) ] CAUSE [BECOME +DRY_01 (y Referent)]

FunGramKB Lexicon

Conceptual Logical Structures

FunGramKB Lexicon

LCM Core Grammar

Aktionsart: Accomplishment
Causative Accomplishment
Variables: x, y
Idiosyncratic features: none
TF Mapping: x = Theme, y = Referent

secar

FunGramKB Lexicon

FunGramKB Ontology

FunGramKB Lexicon

FunGramKB Lexicon

FunGramKB Lexicon

Thematic Frame

(x1) Theme
(x2: +CORPUSCULAR_00) Referent

Meaning Postulate

+(e1: +CHANGE_00 (x1) Theme
(x2) Referent (f1: (e2: n +BECOME_00 (x2) Theme (x3: +WET_00) Attribute)) Result)

Conceptual Logical Structures

BECOME +DRY_01 (y Referent)

[do (x Theme, 0) ] CAUSE [BECOME +DRY_01 (y Referent)]

LCM Core Grammar

Aktionsart: Accomplishment
Causative Accomplishment
Variables: x, y
Idiosyncratic features: none
TF Mapping: x = Theme, y = Referent

FunGramKB Ontology

dry

FunGramKB Lexicon

english

FunGramKB Lexicon

spanish

English

LCM Core Grammar

Aktionsart: Accomplishment
Causative Accomplishment
Variables: x, y
Idiosyncratic features: none
TF Mapping: x = Theme, y = Referent

FunGramKB Lexicon

spanish

secar

CLS Constructor

FunGramKB Ontology
Thematic Frame
(x1) Theme
(x2: +CORPUSCULAR_00) Referent

Meaning Postulate
+(e1: +CHANGE_00 (x1) Theme
x2) Referent (f1: (e2: n +BECOME_00
(x2) Theme (x3: +WET_00) Attribute)) Result)

+DRY_01

Conceptual Logical Structures
BECOME +DRY_01 (y Referent)
[do (x Theme = 0)] CAUSE [BECOME
+DRY_01 (y Referent)]

Aktionsart: Accomplishment
Causative Accomplishment
Variables: x, y
Idiosyncratic features: none
TF Mapping: x = Theme, y = Referent

LCM Core Grammar

FunGramKB Lexicon
Spanish
secar

FunGramKB Ontology

FunGramKB Lexicon
English
dry
Betty asked Bill for an apple.

Meaning #1: To say a question

Meaning #2: To make a request

<IF DECL<br> TNS PAST<br>[do (%BETTY_00 Theme, [+REQUEST_01<br>(%BETTY_00 Theme, %BILL_00 Goal)])] PURP [do (%BILL_00 Goal, 0)]<br>CAUSE [BECOME +REQUEST_01 (%BETTY_00 Theme,<br>+APPLE_00 Referent)]>>>
IF the word is linked to more than one meaning THEN
  IF the word occurs in an idiom/higher-level construction THEN
    Translate the entire phrase with its idiomatic meaning
  ELSE
    Use morphosyntactic constraints
    IF only one meaning is left THEN
      Use the appropriate meaning
    ELSE
      Take into account selectional preferences of arguments
      IF only one meaning is left THEN
        Use the appropriate meaning
      ELSE
        IF the word has already occurred in the text THEN
          Use the same meaning used in its previous occurrence
        ELSE
          IF any of the meanings of the word belongs to the same technical domain as the input text THEN
            Use the appropriate meaning
          ELSE
            Apply a spreading activation method
            IF there is a winning candidate THEN
              Use the appropriate meaning
            ELSE
              Take the most frequent meaning
            END IF
          END IF
        END IF
      END IF
    END IF
  END IF
END IF
The Word-Sense Disambiguation algorithm

IF the word is linked to more than one meaning THEN
    IF the word occurs in an idiom/higher-level construction THEN
        Translate the entire phrase with its idiomatic meaning
    ELSE
        Use morphosyntactic constraints
        IF only one meaning is left THEN
            Use the appropriate meaning
        ELSE
            Take into account selectional preferences of arguments
            IF only one meaning is left THEN
                Use the appropriate meaning
            ELSE
                IF the word has already occurred in the text THEN
                    Use the same meaning used in its previous occurrence
                ELSE
                    IF any of the meanings of the word belongs to the same
technical domain as the input text THEN
                        Use the appropriate meaning
                    ELSE
                        Apply a spreading activation method
                        IF there is a winning candidate THEN
                            Use the appropriate meaning
                        ELSE
                            Take the most frequent meaning
                        END IF
                    END IF
                END IF
            END IF
        END IF
    END IF
ELSE
    Use the appropriate meaning
END IF
The Word-Sense Disambiguation algorithm

IF the word is linked to more than one meaning THEN
  IF the word occurs in an idiom/higher-level construction THEN
    Translate the entire phrase with its idiomatic meaning
  ELSE
    Use morphosyntactic constraints
    IF only one meaning is left THEN
      Use the appropriate meaning
    ELSE
      Take into account selectional preferences of arguments
      IF only one meaning is left THEN
        Use the appropriate meaning
      ELSE
        IF the word has already occurred in the text THEN
          Use the same meaning used in its previous occurrence
        ELSE
          IF any of the meanings of the word belongs to the same technical domain as the input text THEN
            Use the appropriate meaning
          ELSE
            Apply a spreading activation method
            IF there is a winning candidate THEN
              Use the appropriate meaning
            ELSE
              Take the most frequent meaning
            END IF
          END IF
        END IF
      END IF
    END IF
  END IF
END IF

Thank you!

E-mail: jcperinan@pdi.ucam.edu