# The Unit Graphs Framework: Foundational Concepts and Semantic Consequence

**Maxime Lefrançois & Fabien Gandon**

**Research question:** how to reason over linguistic knowledge of the Explanatory and Combinatorial Dictionary ?
**Proposed approach:** provide Unit Graphs with Model Semantics based on Relational Algebra.

## Hierarchy of Unit Types

$$\mathcal{T} \stackrel{\text{def}}{=} (T_D, \boldsymbol{S}_{\mathcal{T}}, \boldsymbol{\gamma}, \boldsymbol{\gamma_1}, \boldsymbol{\gamma_0}, C_A, \perp^{\sqcap}_A, \{\varsigma_t\}_{t \in \boldsymbol{T}})$$

**Unit Types** are described in the Lexicon
vs. **Units** are described in Sentence Representations

Deep Semantics ⇔ Surface Semantics ⇔ Deep Syntax ⇔ Surface Syntax ⇔ *Texts*
*Meanings*

Linguistic Units Types used at these levels have an **Actantial Structure**
**Actant Slots** (ASlots) may be optional, obligatory, *and prohibited*
In the Hierarchy of Unit Types, a Unit Type **inherits** and may **specialize** the Actantial Structure of its parents

### Actant Symbols (ASymbols)

$S_{\mathcal{T}}$ Deep Semantic: Lexicalized Semantic Roles
Surface Semantics: Numbers
Deep Syntax: Roman numerals
...

### Primitive Unit Types (PUTs)

$\boldsymbol{T}$ is the disjoint union of:

- the set of declared PUTs $T_D$
  ex: Lexical unit type ANIMAL
  Grammatical unit type *Verb*, *Noun*, *plur*
  Surface Semantic unit type ⁽animal⁾
  Deep Semantic unit type /animal\
- radices *(the roots)* $\gamma(S_{\mathcal{T}})$
- obligant (those that make obligatory) $\gamma_1(S_{\mathcal{T}})$
- prohibent (those that prohibit) $\gamma_0(S_{\mathcal{T}})$
- The prime absurd PUT $\perp$
- The prime universal PUTs $\top$

### Pre-order over PUTs

$C_A$ is used to compute a pre-order over PUTs, and to assign a set of ASlots to each PUT.
ex: X eats Y (in Z) ($\gamma_1$(1), ⁽eat⁾), ($\gamma_1$(2), ⁽eat⁾), ($\gamma$(3), ⁽eat⁾)
($\gamma$(subject), *Verb*)
(*pluralizable*, *plur*)
(/animal\ ,/dog\)

### Signatures of ASlots

$\{\varsigma_t\}_{t \in \boldsymbol{T}}$ $\varsigma_t(s)$ denotes the type of units that fill ASlot $s$ of a unit of type $t$

ex: $\varsigma_{(\text{to eat})}(1) = \{(\text{animal})\}$

### The Unit Types Actantial Structure

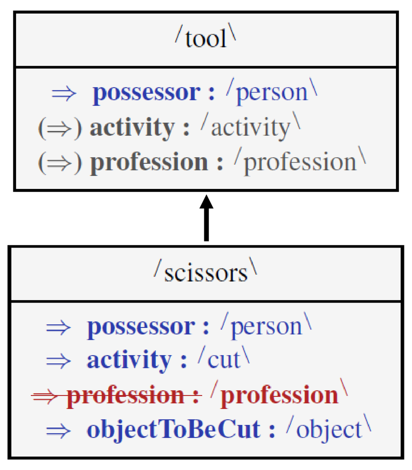
is the set of its obligatory, prohibited, and optional ASlots, and their signature

A Unit may consist of several conjoint PUTs

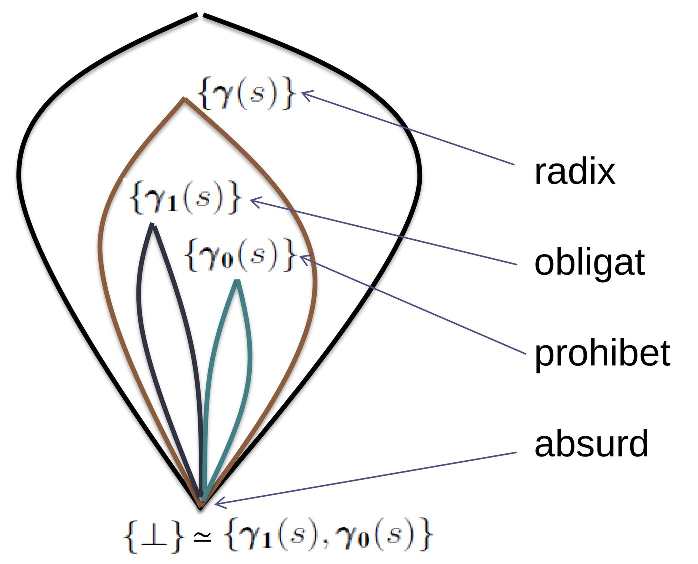
### Conjunctive Unit Types (CUTs)

ex: { *def*, *plur*, ANIMAL } (⁽the animals⁾)

- the actantial structure of PUTs is naturally extended to CUTs
- some CUTs $\perp^{\sqcap}_A$ are asserted to be absurd.
- the pre-order over PUTs is extended to a pre-order over CUTs
- absurd CUTs are those lower than the prime absurd PUT $\{\perp\}$




*Example of inheritance and specialization in the hierarchy of Deep Semantic Unit Types*




*Organization of the Unit Types Hierarchy with respect to a unique ASymbol s*
*The complete Unit Types Hierarchy is an intricated superposition of such figures*

## Hrchy of Circumstantial Symbols

$$\mathcal{C} \stackrel{\text{def}}{=} (\boldsymbol{S}_{\mathcal{C}}, \boldsymbol{C}_{\boldsymbol{S}_{\mathcal{C}}}, \mathcal{T}, \{\boldsymbol{\sigma}_s\}_{s \in \boldsymbol{S}_{\mathcal{C}}})$$

### Circumstantial Symbols (CSymbols)

$S_{\mathcal{C}}$ Deep and Surface Semantics: none
Deep Syntax: **ATTR**, **COORD**, **APPEND**
...

### Asserted CSymbols Comparisons

$C_{S_{\mathcal{C}}}$ is used to compute a pre-order over CSymbols
ex: (deep syntactic CSymbol, **ATTR**)

### Signatures of CSymbols

$\{(domain(s), range(s))\}_{s \in S_{\mathcal{C}}}$
it denotes the type of units that a circumstantial relation with symbol s links

## Unit Graphs

are defined over a Support $\mathcal{S} \stackrel{\text{def}}{=} (\mathcal{T}, \mathcal{C}, \boldsymbol{M})$

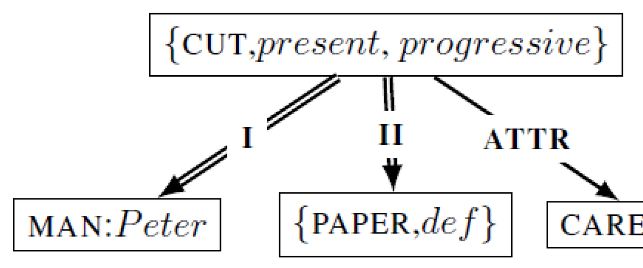
### Unit Node Markers

$\boldsymbol{M}$ *Arbitrary Symbols*
*Every Element of **M** identifies a specific unit;*
*Multiple elements of **M** may identify the same unit.*

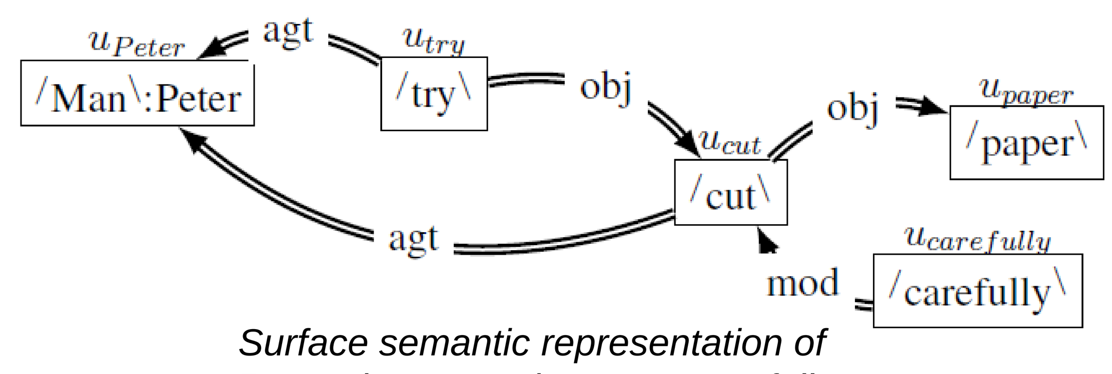
### Unit Graphs

$$G \stackrel{\text{def}}{=} (U, \boldsymbol{l}, A, C, Eq)$$

- Unit nodes
- Unit nodes labels : a type + a marker
- Actantial triples
- Circumstantial triples
- Declared equivalences of unit nodes




*Deep syntactic representation of*
Peter is carefully cutting the paper




*Surface semantic representation of*
Peter tries to cut the paper carefully

## Model of a Support $\mathcal{S}$ :

$$M = (D, \delta)$$

the **domain** contains:
- a special ● element, that represents *nothing*,
- *plus at least one more element*

the **interpretation function** must be such that:
- $M$ is a model of $\mathcal{T}$;
- $M$ is a model of $\mathcal{C}$;
- $\forall m \in \boldsymbol{M}, \delta(m) \in D \setminus \bullet$;

## Model of a Hrchy of Unit Types

The interpretation $\delta(\{t\})$of a PUT *t* is a relation of arity: *1 + its number of ASlots*, with attributes 0 and the symbols of its ASlots

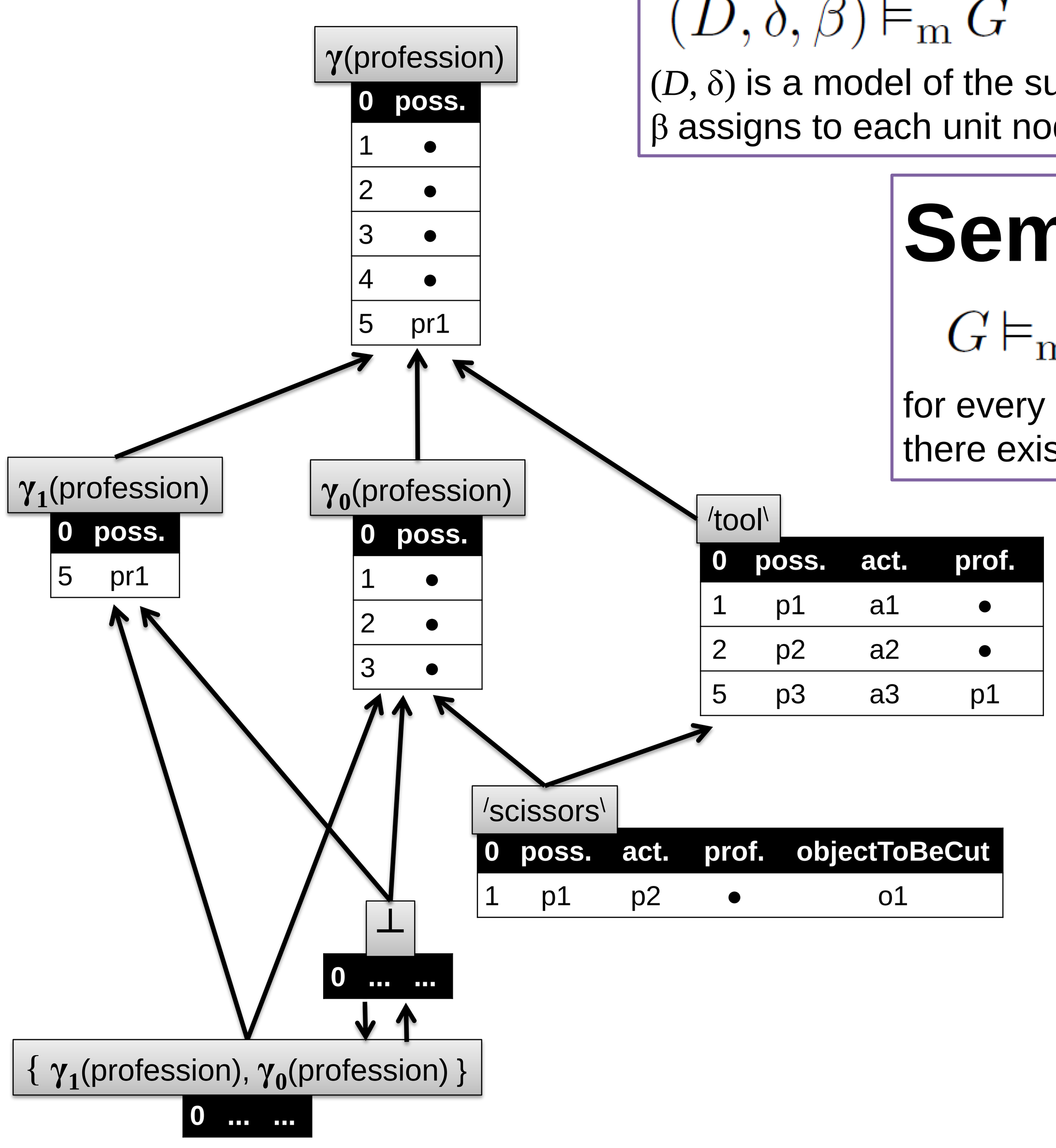
In a row:
- Element 0 is the unit itself
- Element *s* is the unit that fills ASlot *s*
- If the element *s* is ● then *there is no unit* that fills ASlot *s*

We defined equations that $\delta(\{t\})$must satisfy, and extend the interpretation function to CUTs

## Model of a Hrchy of CSymbols

The interpretation $\delta(s)$ of a CSymbol $s$ is a binary relation, with attributes *gov* (the governor) and *circ* (the circumstant)

We defined equations that $\delta(s)$ must satisfy




*Example of a model of a Unit Types hierarchy*

## Model satisfying a Unit Graph

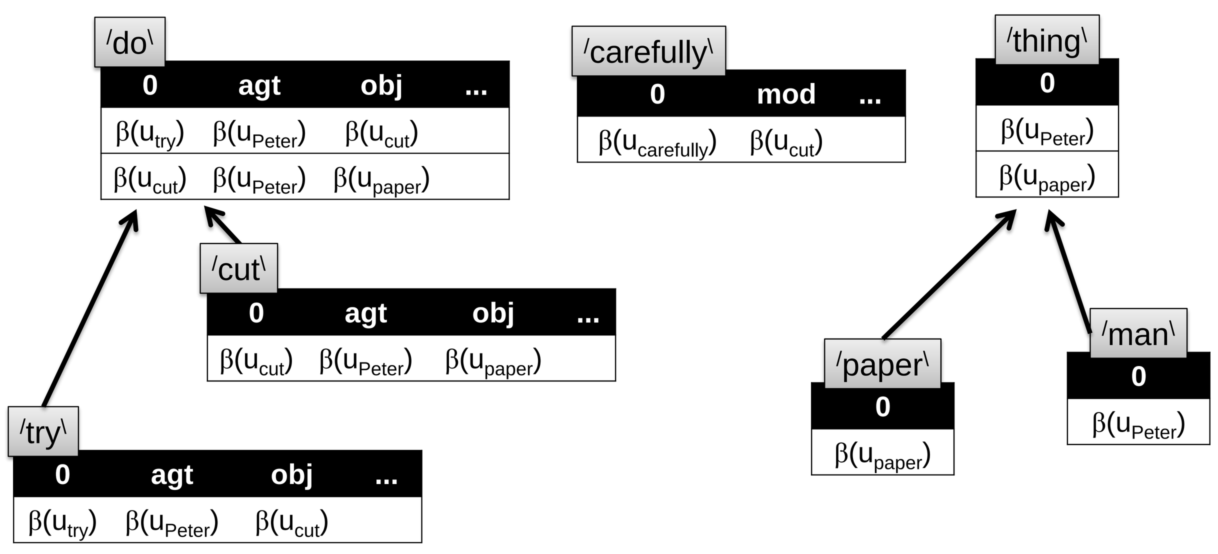
$$(D, \delta, \beta) \vDash_{\text{m}} G$$

($D$, δ) is a model of the support
β assigns to each unit node an element of $D$, and must satisfy some equations

## Semantic Consequence

$G \vDash_{\text{m}} H$ iif

for every model and for any assignment $\beta_G$ such that $(D, \delta, \beta_G) \vDash_{\text{m}} G$ ,
there exists an assignment $\beta_H$ such that $(D, \delta, \beta_H) \vDash_{\text{m}} H$




*Example of a minimal model satisfying Unit Graph **G** :*
*the Deep syntactic representation of*
Peter is carefully cutting the paper

## Future Work:

- include Definitions of PUTs (they represent formal Lexicographic Definitions for instance) in the support
- define algorithms to complete a domain, and to check semantic consequence
- find conditions on the unit type hierarchy and the definitions to ensure such algorithms are decidable

Wimmics, Inria, I3S, UNSA, CNRS