XMG: Using principles

Laura Kallmeyer & Simon Petitjean

HHU Düsseldorf

WS 2015

15.01.2016

Principles: motivation

- As fragments become more numerous, controlling their combination (and the scope of variables) gets difficult
- Idea: adding new constraints on top of dominance and precedence
- Principles: sets of additionnal constraints for the solver, introduced in [Crabbé and Duchier, 2004]

A set of principles

XMG offers several sets of additionnal constraints over the models (principles):

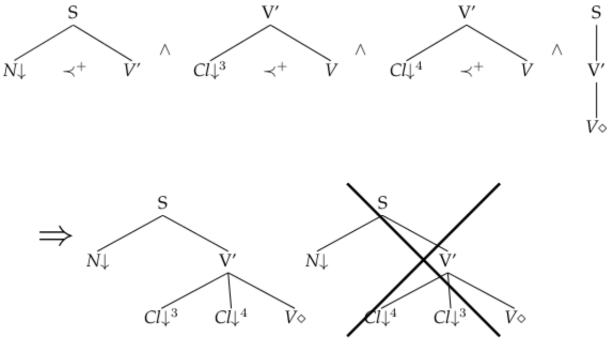
- colors: polarities for node unification
- rank: linear order constraints on nodes
- unicity: uniqueness of a feature inside a model

Rank: Clitics ordering

- The ordering of clitic pronouns (in Spanish or French for example) is known to be problematic when formalizing a grammar
- In a metagrammar, when combining fragments, nodes representing these clitics have to come in a specific order

- Pedro nos la da
- *Pedro la nos da
- Je le lui laisse
- *Je lui le laisse

Rank: Clitics ordering (in French)



- Every produced model has to satisfy the order constraint

Using principles: rank

```
use rank with () dims (syn)
type RANK=[1..7]
property rank: RANK
```

```
class CliticIobjectII
import nonReflexiveClitic[]
{
  <syn>{
    node xCl(rank=2)
            [top=[func=iobj, pers = @{1,2}]]
  }
}
```

Using principles: unicity

```
use unicity with (rank=1) dims (syn)
use unicity with (rank=2) dims (syn)
use unicity with (rank=3) dims (syn)
use unicity with (rank=4) dims (syn)
use unicity with (rank=5) dims (syn)
use unicity with (rank=6) dims (syn)
use unicity with (rank=7) dims (syn)
```

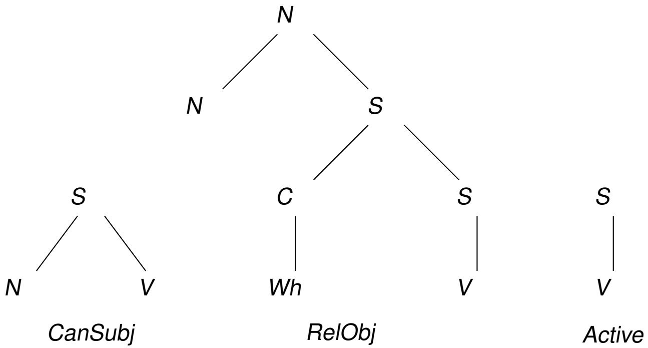
Using principles: colors

- Colors are a solution to guide the combination of fragments
- A color is affected to every node
- New constraints on node unification

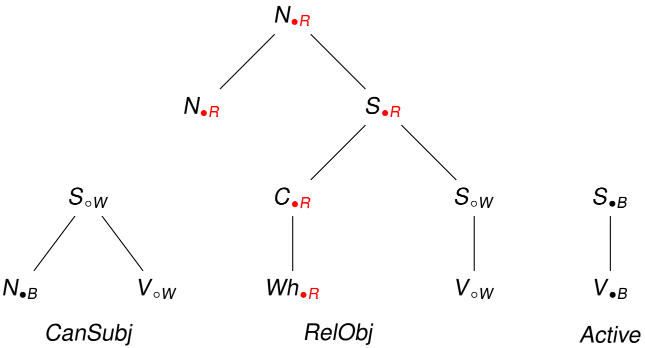
	$\bullet_B$	$\bullet_R$	$\circ_W$	$\perp$
$\bullet_B$	$\perp$	$\perp$	$\bullet_B$	$\perp$
$\bullet_R$	$\perp$	$\perp$	$\perp$	$\perp$
$\circ_W$	$\bullet_B$	$\perp$	$\circ_W$	$\perp$
$\perp$	$\perp$	$\perp$	$\perp$	$\perp$

- Valid models only have red and black nodes

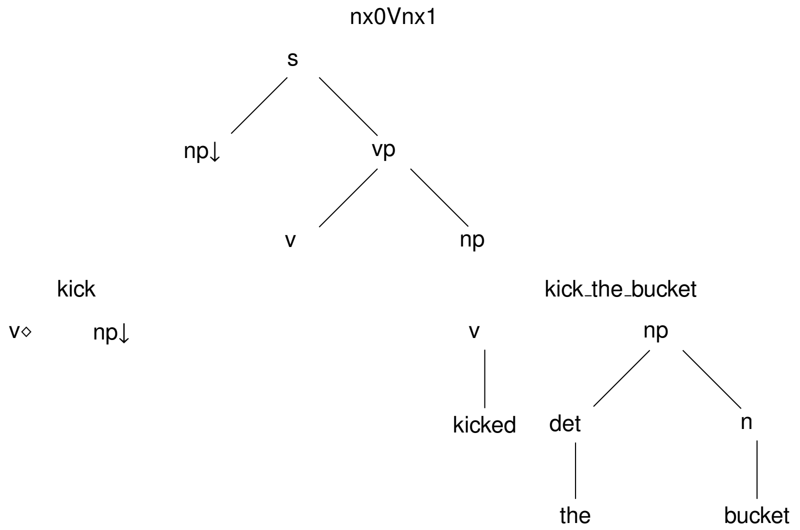
Combination with polarities



Combination with polarities

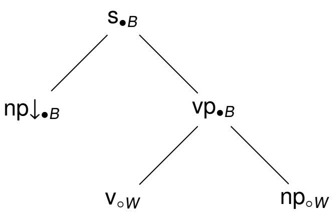


Combination with polarities



Combination with polarities

nx0Vnx1

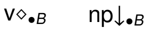


$s_{\bullet B}$

$np{\downarrow}_{\bullet B}$ $vp_{\bullet B}$

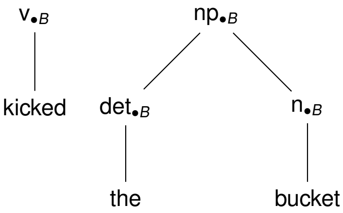
$v_{\circ W}$ $np_{\circ W}$

kick



$v{\diamond}_{\bullet B}$ $np{\downarrow}_{\bullet B}$

kick_the_bucket



$v_{\bullet B}$ $np_{\bullet B}$

kicked $det_{\bullet B}$ $n_{\bullet B}$

the bucket

Using principles: colors

```
use color with () dims (syn)
type COLOR={red,black,white}
property color: COLOR
```

```
class nx0Vnx1
declare ?S ?NP_Subj ?VP ?V ?NP_Obj
{
  <syn>{
     ?S (color=red)[cat=s] {
       ?NP_Subj (color=black, mark=subst) [cat=np]
       ?VP (color=black)[cat=vp] {
         ?V (color=white)[cat=v]
         ?NP_Obj (color=white)[cat=np]
       }
     }
  }
}
```

Exercise

Adapt the existing toy metagrammar of verbs

- Every node must get a color
- The set of models should remain identical
- You may have to decide again which variables should be exported or not
- You can use the file `Metagrammars/synsem/Misc/Verbs/verbs4_colors_TODO.mg` (after a `bzr pull`)

Crabbé, B. and Duchier, D. (2004).
Metagrammar Redux.
In *International Workshop on Constraint Solving and Language Processing*, Copenhagen.