Iterated Dependencies and Kleene Iteration

Michael Dekhtyar, Alexander Dikovsky and Boris Karlov

LINA CNRS UMR 6241, Université de Nantes (France) and Tver State University (Russia)

Plan

- 1. Dependency syntax and d-structures
- 2. Categorial dependency grammars
- 3. Kleene star closure problem
  - 3.1. A solution using multi-modal CDG
  - 3.2. Solution complexity
- 4. Conclusion

1. Dependency syntax and d-structures

Dependency syntax

The traditional grammar is *word-oriented* [W.K.Percival'90]. It defines the syntax in terms of *dependencies*: relations between words.

First formalized approach to dependency structures is due to L.Tesnière [1934].
Syntactic structure is defined in terms of two kinds of binary relations on words:
- linear precedence order of words in the sentence, and
- syntactic dependencies whose graph is acyclic and rooted.

$c-copul$, $pred$, det, $prepos-in$ are dependency names.

Formal dependency grammars have appeared in the early 60ies [D.Hays'61, E.Gaifman'62,65]

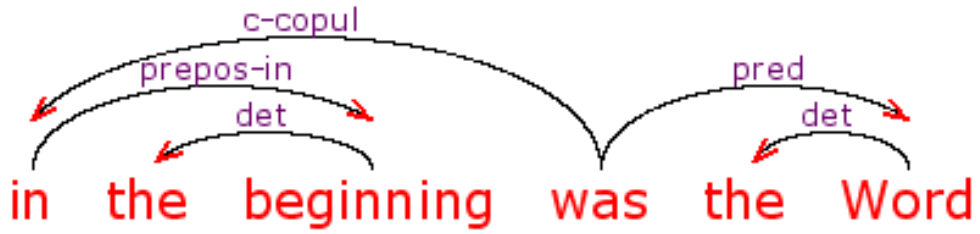
Dependency structures

Dependency relations are asymmetric: $W_1 \xrightarrow{d} W_2$:

W_1 is a governor (head) of W_2 through dependency d

W_2 is a subordinate of W_1 through d.

Dependency structure (DS) of a sentence x is linearly ordered by the precedence order in x.



$was \xrightarrow{pred} Word$; $was \xrightarrow{c-copul} in$

Dominance $W_1 \rightarrow^* W_2$:

reflexive-transitive closure of dependency relations: $\exists d(W_1 \xrightarrow{d} W_2)$

$was \rightarrow^* the$; $Word \rightarrow^* Word$.

Projection $Proj(W) = \{W' \parallel W \rightarrow^* W'\}$ ordered by precedence
(a generalized *constituent* with the *head* W).

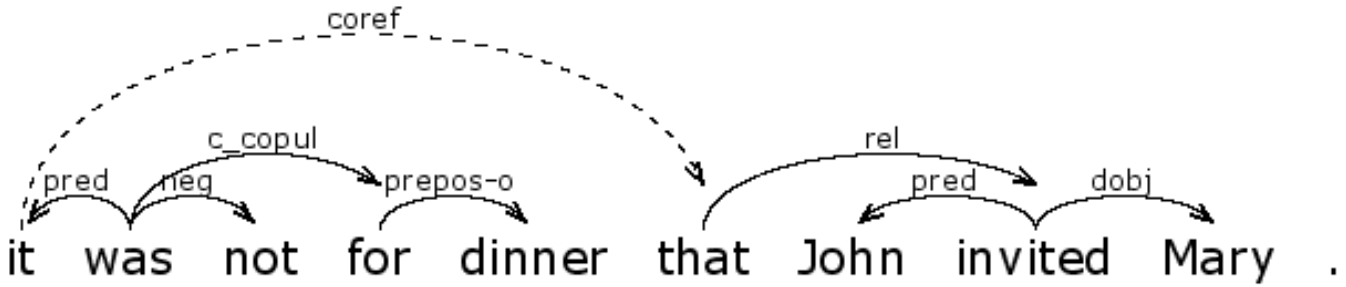
Projective DS

Projective DS: words' projections are intervals. So the dependencies don't cross 

In all languages, the majority of DS are projective

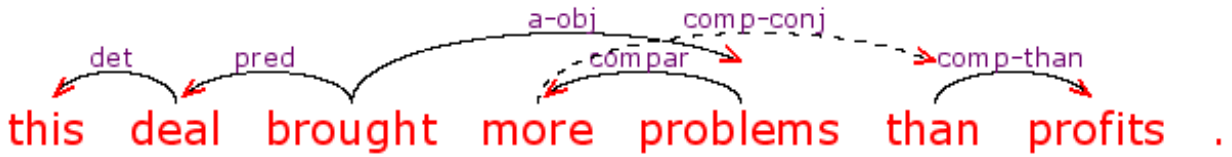
Many authors of parsers assume the projectivity. Not the linguists. . .

Non-projective dependencies are caused by:
- discontinuous constructions and
- fronting lexical units which change their communicative status (e.g. are topicalized or defocalized).



Non-projective DS

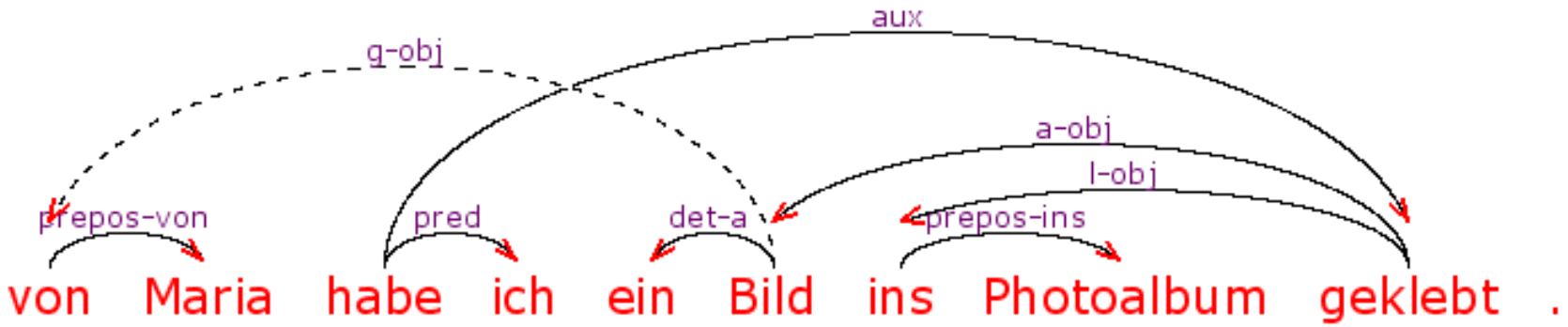
There are regular non-projective constructions in many languages.
Discontinuous comparatives in English:



Negation, comparatives, clitics, etc. in French:

(Fr. $^*She\ it_{g=f}\ him\ has\ given$)

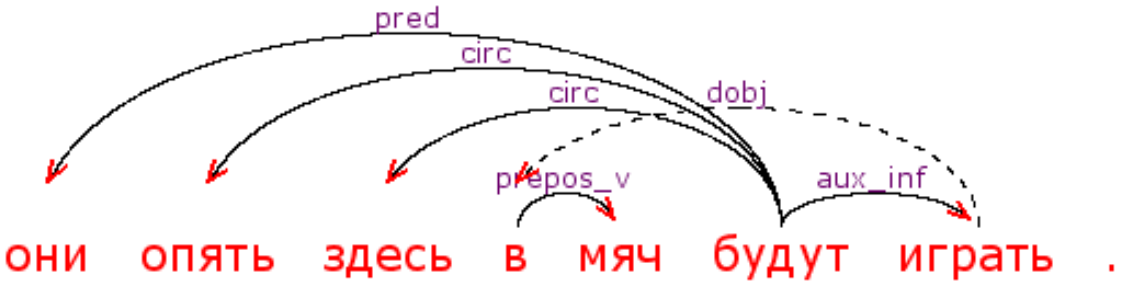
Scrambling in German:



(Gr. $^*of,\ Maria\ have\ I\ a\ photo\ in\ the\ album\ sticked\ on$)

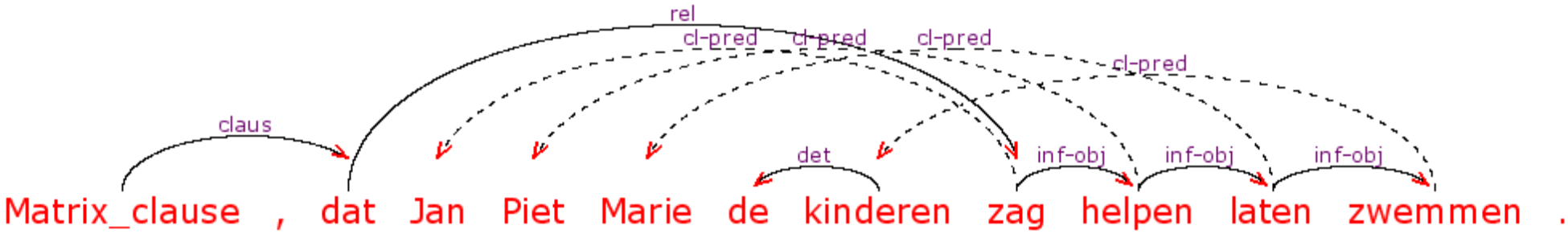
Complex non-projective DS

Flexible word order in Slavonic languages:



(Rus. *they again here the ball will play*)

Cross-serial dependencies in Dutch:



(Dutch * ... *that Jan Piet Marie children saw help teach swim*)

CONCLUSION: Dependency grammars must define non-projective DS.

2. Categorial dependency grammars

- are type logical grammars (categorial grammars with FO types),
- define unlimited DS (including discontinuous and long distance dependencies),
- are lexicalized and completely local (grammars of words' valencies),
- are adapted to flexible order (extended by flat regular expressions),
- are polynomially parsed (efficient deterministic offline parser is implemented),
- are well suited for practical use (cf. a wide scope large scale CDG of French).

Types of projective dependencies

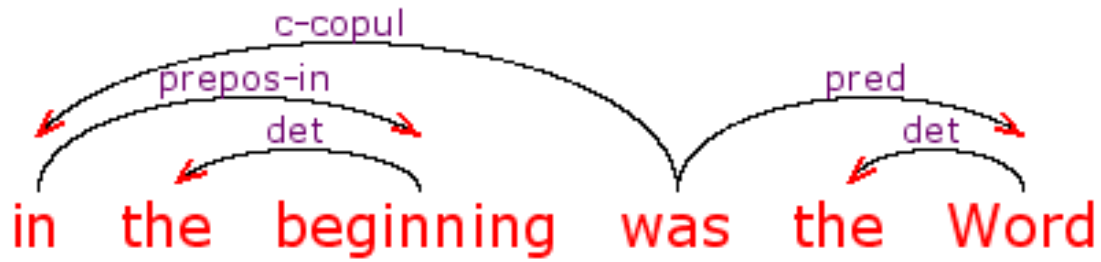
$Gov \xrightarrow{d} Sub$ is typed as :

$Gov \mapsto [..\backslash../../d/..]$ for the governor

and as

$Sub \mapsto [..\backslash d/..]$ for the subordinate.

EX:



$in \mapsto [c\_copul/prepos-in]$ $the \mapsto [det]$ $beginning \mapsto [det\backslash prepos-in]$

$was \mapsto [c\_copul\backslash S/pred]$ $Word \mapsto [det\backslash pred]$

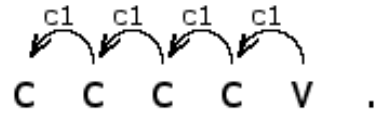
Necessity of iterated types

Specific interpretation of subtype elimination:

$a \mapsto [\alpha\backslash d]$, $b \mapsto [d\backslash\beta]$ derives the dependency $a \overset{d}{\longleftarrow} b$ for ab

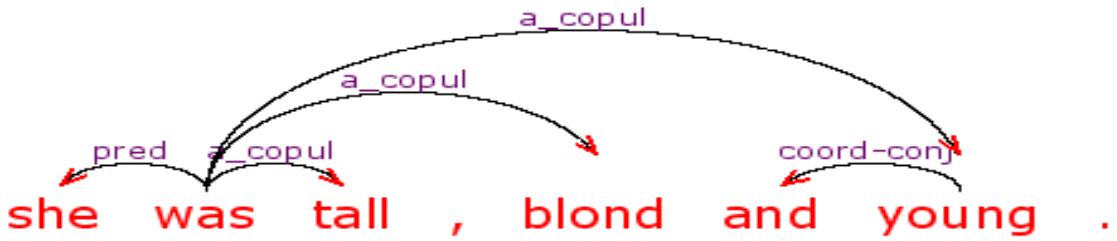
So the recursive types derive sequenced dependencies:

$v \mapsto [c1\backslash S]$, $c \mapsto [c1\backslash c1], [c1]$ derives for $ccccv$ the DS:



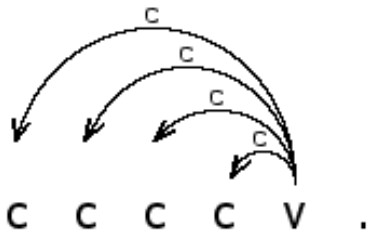
Principle of optional repeatable dependencies (see [Mel'cuk'88]):

- modifiers of a noun n share the governor n,
- circumstantials of a verb v share the governor v.



This is why the explicit iterated types are needed:

$v \mapsto [c*\backslash S]$, $c \mapsto [c]$ derives for $ccccv$ the DS:



Types of non-projective dependencies

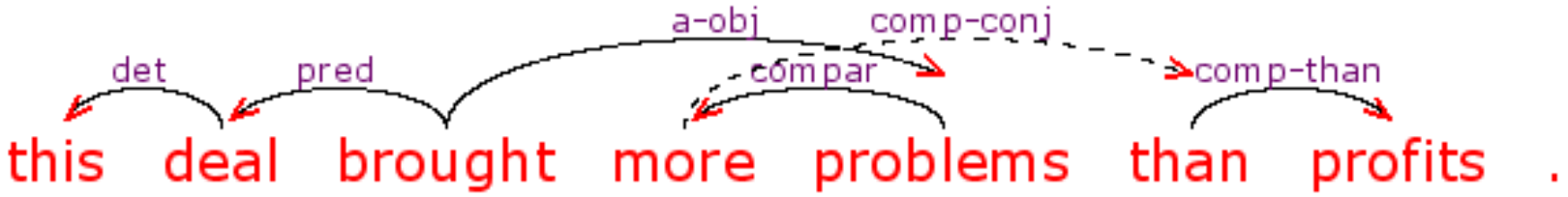
are defined using dual polarized valencies:

$Gov -\!\!\overset{d}{-}\!\!> Sub$:

$Gov \mapsto [..]^{\nearrow d}$ for the governor

and

$Sub \mapsto [..]^{\searrow d}$ for the subordinate.



$this \mapsto [det]$ $deal \mapsto [det\backslash pred]$ $brought \mapsto [pred\backslash S/a\!-\!obj]$
$problems \mapsto [compar\backslash a\!-\!obj]$ $profits \mapsto [comp\!-\!than]$
$more \mapsto [compar]^{\nearrow comp\text{–}conj}$ $than \mapsto [/comp\!-\!than]^{\searrow comp\text{–}conj}$

Anchoring distant subordinates

Host words may anchor distant subordinates using anchor types :

$$Host\ word \mapsto [..\backslash../../\#(\searrow d)/..]$$

$$Anchored\ word \mapsto [..\backslash\#(\searrow d)/..]^{\searrow d}$$

(Fr. $^{*}She\ it_{g=f}\ him\ has\ given$)

elle $\mapsto [pred]$

la $\mapsto [\#(\swarrow clit-a-obj)]^{\swarrow clit\text{–}a\text{–}obj}$

lui $\mapsto [\#(\swarrow clit-3d-obj)]^{\swarrow clit\text{–}3d\text{–}obj}$

a $\mapsto [\#(\swarrow clit-3d-obj)\backslash\#(\swarrow clit-a-obj)\backslash pred\backslash S/aux]$

donnée $\mapsto [aux]^{\nwarrow clit\text{–}3d\text{–}obj \nwarrow clit\text{–}a\text{–}obj}$

CDG calculus

$\mathbf{L^l}$. $C^{P_1}[C\backslash\beta]^{P_2} \vdash [\beta]^{P_1P_2}$

$\mathbf{I^l}$. $C^{P_1}[C^*\backslash\beta]^{P_2} \vdash [C^*\backslash\beta]^{P_1P_2}$

$\mathbf{\Omega^l}$. $[C^*\backslash\beta]^{P} \vdash [\beta]^{P}$

$\mathbf{D^l}$. $\alpha^{P_1(\swarrow C)P(\nwarrow C)P_2} \vdash \alpha^{P_1PP_2}$,

if $(\swarrow C)P(\nwarrow C)$ satisfies the following valency pairing principle:

FA: in $(\swarrow C)P(\nwarrow C)$, the valency $\swarrow C$ is the **first available** for the dual valency $\nwarrow C$, i.e. P has no occurrences of $\swarrow C, \nwarrow C$.
Similar for right valencies.

Elimination of dual valencies $v = (\swarrow C)$, $\breve{v} = (\nwarrow C)$ by the rule $\mathbf{D^l}$ creates the *discontinuous dependency* C.

Elimination of anchored valency $\#(v)$ by the rules $\mathbf{L^l}, \mathbf{I^l}$ creates *not a dependency*, but an anchor link.

Proofs of typing correctness

EX: *Elle la lui a donnée*

Without anchoring (clitics may permute):

$$[pred]\ [\varepsilon]^{\swarrow clit\text{–}dobj}\ [\varepsilon]^{\swarrow clit\text{–}iobj}\ [pred\backslash S/aux]\ [aux]^{\nwarrow clit\text{–}iobj,\nwarrow clit\text{–}dobj}$$

$$\dfrac{[pred]\quad \dfrac{[\varepsilon]^{\swarrow clit\text{–}dobj}\quad \dfrac{[\varepsilon]^{\swarrow clit\text{–}iobj}\quad [pred\backslash S/aux]}{[pred\backslash S/aux]^{\swarrow clit\text{–}iobj}}(\mathbf{L}^l)}{[pred\backslash S/aux]^{\swarrow clit\text{–}dobj\swarrow clit\text{–}iobj}}(\mathbf{L}^l)}{\dfrac{[S/aux]^{\swarrow clit\text{–}dobj\swarrow clit\text{–}iobj}\quad [aux]^{\nwarrow clit\text{–}iobj\nwarrow clit\text{–}dobj}}{[S]^{\swarrow clit\text{–}dobj\swarrow clit\text{–}iobj\nwarrow clit\text{–}iobj\nwarrow clit\text{–}dobj}}(\mathbf{L}^r)}(\mathbf{L}^l)}{S}(\mathbf{D}^l\times 2)$$

With anchoring (clitics cannot permute):

$$[pred]\ [\#^l(\swarrow clit-dobj)]^{\swarrow clit\text{–}dobj}\ [\#^l(\swarrow clit-iobj)]^{\swarrow clit\text{–}iobj}\ [\#^l(\swarrow clit-iobj)\backslash\#^l(\swarrow clit-dobj)\backslash pred\backslash S/aux]\ [aux]^{\nwarrow clit\text{–}iobj,\nwarrow clit\text{–}dobj}$$

$$\dfrac{[pred]\quad \dfrac{[\#^l(\swarrow clit-dobj)]^{\swarrow clit\text{–}dobj}\quad \dfrac{[\#^l(\swarrow clit-iobj)]^{\swarrow clit\text{–}iobj}\quad [\#^l(\swarrow clit-iobj)\backslash\#^l(\swarrow clit-dobj)\backslash pred\backslash S/aux]}{[\#^l(\swarrow clit-dobj)\backslash pred\backslash S/aux]^{\swarrow clit\text{–}iobj}}(\mathbf{L}^l)}{[pred\backslash S/aux]^{\swarrow clit\text{–}dobj\swarrow clit\text{–}iobj}}(\mathbf{L}^l)}{\dfrac{[S/aux]^{\swarrow clit\text{–}dobj\swarrow clit\text{–}iobj}\quad [aux]^{\nwarrow clit\text{–}iobj\nwarrow clit\text{–}dobj}}{[S]^{\swarrow clit\text{–}dobj\swarrow clit\text{–}iobj\nwarrow clit\text{–}iobj\nwarrow clit\text{–}dobj}}(\mathbf{L}^r)}(\mathbf{L}^l)}{S}(\mathbf{D}^l\times 2)$$

CDG Definition

CDG $G = (W, \mathbf{Cat}, S, \lambda)$:

Dictionary: W

Types: the set of (polarized) categories $\mathbf{Cat}$, $S \in \mathbf{Cat}$ (axiom)

Lexicon (type assignment): $\lambda : W \to 2^{\mathbf{Cat}}$ (i.e. $w \mapsto \{c_1, ..., c_k\} \subset \mathbf{Cat}$).

Calculus (one for all CDG): $\alpha_1 ... \alpha_n \vdash \alpha, \quad \alpha, \alpha_1, ..., \alpha_n \in \mathbf{Cat}$

Language: $L(G) = \{x \in W^+ \mid (\exists\, \Gamma)\ (\Gamma \in \lambda(x) \ \ \wedge \ \ \Gamma \vdash^* S)\}$

DS-language: $D(G) =$
$\{D(\pi) \mid (\exists\, x)(\exists\, \Gamma)\ (x \in L(G) \ \ \wedge \ \ \Gamma \in \lambda(x) \ \ \wedge \ \ \pi = \Gamma \vdash^* S)\}$

EX: $L(G_{abc}) = \{a^n b^n c^n \mid n > 0\}$ for the CDG G_{abc} :
$a \mapsto A^{\swarrow A},\ [A \backslash A]^{\swarrow A},$
$b \mapsto [B/C]^{\nwarrow A},\ [A \backslash S/C]^{\nwarrow A},$
$c \mapsto C,\ [B \backslash C]$

Properties of CDG

Projections. *local* : $\|C^P\|_l = C$ and *valency* : $\|C^P\|_v = P$

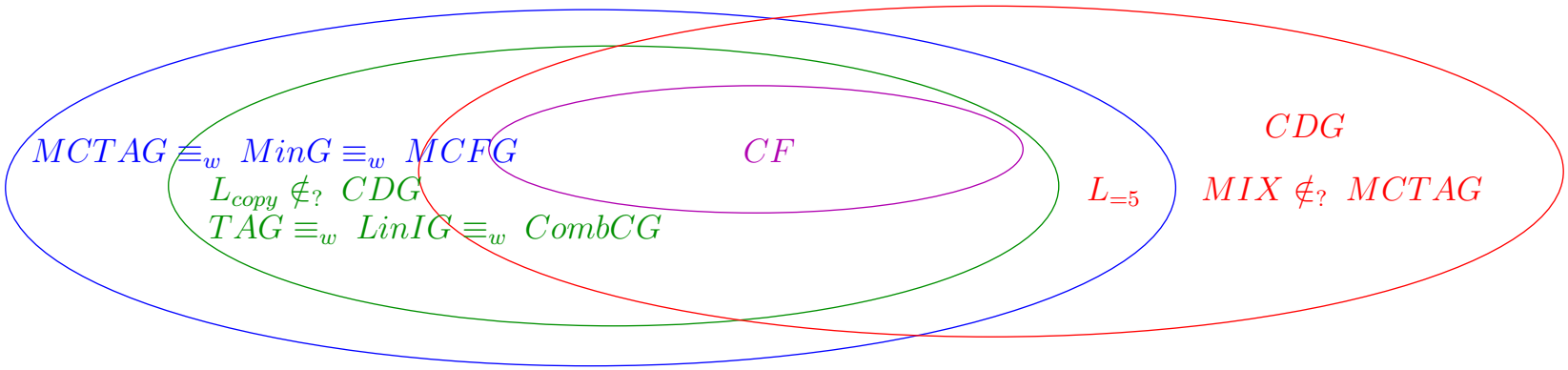
Projective dependency calculus $\vdash_{\mathbf{c}}$: rules $\mathbf{L^l}$, $\mathbf{I^l}$, $\mathbf{\Omega^l}$.

TH: For a CDG $G = (W, \mathbf{C}, S, \lambda)$ and $x \in W^+$,
$x \in L(G)$ iff there is $\Gamma \in \lambda(x)$ such that:

1. $\|\Gamma\|_l \vdash_{\mathbf{c}} S$,
2. $\|\Gamma\|_v$ is well-bracketed for every valency.

Parsing complexity (Dekhtyar,Dikovsky'2004,2008) :

1. theoretical : $\mathbf{O(n^{3+2p})}$ (for p polarized valencies)
2. in practice: $\mathbf{O(n^4)}$ ($\mathbf{O(n^3)}$ for projective dependencies).



$L_{copy} = \{ww \mid w \in W^+\}$, $L_{=i} = \{a_1^n \dots a_i^n \mid n > 0\}$, $MIX = \{w \in \{a,b,c\}^+ \mid |w|_a = |w|_b = |w|_c\}$

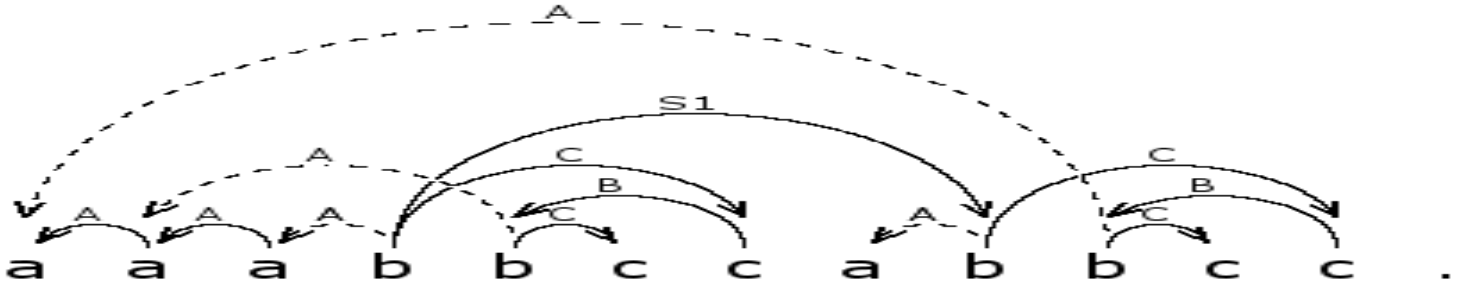
3. Kleene star closure problem

Closure under operations

$\mathcal{L}(CDG)$ is closed under HOM, HOM^{-1}, $\cap REG$, $\cup$, $\circ$ ([Dekhtyar,Dikovsky'2008])

Explicit iterated types may suggest the idea that $\mathcal{L}(CDG)$ is also closed under iteration * ([Dekhtyar,Dikovsky'2008] without proof). BUT:

EX: let us consider the CDG G:
$a \mapsto A^{\swarrow A},\ [A\backslash A]^{\swarrow A},$
$b \mapsto [B/C]^{\nwarrow A},\ [A\backslash S1/C]^{\nwarrow A},\ [A\backslash S/S1/C]^{\nwarrow A},$
$c \mapsto C,\ [B\backslash C]$
It may seem that $L(G) = L(G_{abc})L(G_{abc})$, but it is not so because it contains, for example, the string $aaabbccabbcc$ which has the DS:



So the straightforward construction does not work!
This is the effect of the completely local definition of discontinuous dependencies.

Main Question

How to limit the scope of a polarized valency?

In particular,

how to fix limits impenetrable for discontinuous dependencies?

E.g., in the preceding example, how to limit the scope of the valency $\nwarrow A$ in $b \mapsto [B/C]^{\nwarrow A},\ [A\backslash S1/C]^{\nwarrow A}$ to the substring $aabbcc$ $aabbcc$ subordinate through S_1 to the first b ?

A multimodal solution

Multimodal CDG ([Dikovsky'07]):

Every polarized valency v may have its own pairing rule $\mathbf{D}_{\mathbf{R_v}}$.

For our problem, we propose the following negative list modality:
Let C be the name of a polarized valency and $\pi(C)$ be a list of polarized valencies.

$$\mathbf{D}_{\mathbf{FA}^{l}_{C:\pi(C)}} \quad \alpha^{P_1(\swarrow C)P(\nwarrow C)P_2} \vdash \alpha^{P_1PP_2},$$

$(\swarrow C)\ P\ (\nwarrow C)$ satisfies the pairing rule
$\mathbf{FA}_{C:\pi(C)}$: P has no occurrences of $\swarrow A, \nwarrow A, \nearrow A, \searrow A$ for $A \in \pi(C)$ and also of $\swarrow C, \nwarrow C$.
Intuitively: the discontinuous dependency C cannot cross discontinuous dependencies in the list $\pi(C)$.

$\mathcal{L}(mmCDG^{-FA})$: class of languages generated by CDG with negative list modalities.

Theorem 1 $\mathcal{L}(mmCDG^{-FA})$ *is closed under Kleene iteration (so it is an AFL).*

Solution complexity

$mmCDG^{-FA}$ turn out to be very expressive:

$L_{exp} = \{1010^2 1 \ldots 10^{2^n} 1 \parallel n > 1\}$ is generated by $mmCDG^{-FA}$ G_{exp}:

$1 \mapsto [S/A_1]^{\nearrow X}, [D_1/C]^{\nearrow Y}, [B/A]^{\searrow X \nearrow X}, [D/C]^{\searrow Y \nearrow Y}, [B/A_2]^{\searrow X \nearrow X},$
$[D/C_2]^{\searrow Y \nearrow Y}, [A_2]^{\searrow X \searrow Y}, [C_2]^{\searrow X \searrow Y}$

$0 \mapsto [A_1/D_1]^{\nearrow B \nearrow B}, [A/A]^{\searrow A \nearrow B \nearrow B}, [A/D]^{\searrow A \nearrow B \nearrow B}, [C/C]^{\searrow B \nearrow A \nearrow A},$
$[C/B]^{\searrow B \nearrow A \nearrow A}, [A_2/A_2]^{\searrow A}, [C_2/C_2]^{\searrow B}$

with $\pi(X) = \{A\}, \pi(Y) = \{B\}$.

Corollary 1 *Languages in $\mathcal{L}(mmCDG^{-FA})$ may be not semilinear.*

Theorem 2 *Membership problem for $mmCDG^{-FA}$ is NP-complete.*

(By reduction of 3-SAT; four discontinuous non-crossing dependencies are used: $\pi(0) = \{1\}$, $\pi(1) = \{0\}$, $\pi(0') = \{1'\}$, $\pi(1') = \{0'\}$.)

$\{ww^R w \parallel\ w \in \{a, b\}^+\}$ and L_{copy} also belong to $\mathcal{L}(mmCDG^{-FA})$.

4. Conclusion

- 1. Categorial Dependency Grammars
  - define unlimited dependency structures (projective, discontinuous, non-tree)
  - use explicit iterated types
  - do not use the non-cross modalities
  - are perfectly adapted for real applications
  - are polynomially analyzed
- 2. One non-crossed discontinuous dependency is sufficient for Kleene iteration closure
- 3. Four non-crossed discontinuous dependencies are sufficient for NP-hardness
- 4. Still remain open the problems:
  - $L_{copy} \notin \mathcal{L}(CDG)$
  - CDG-languages are semilinear
  - closure/non-closure of $\mathcal{L}(CDG)$ under Kleene iteration

$\{a^n b^n c^n \mid n > 0\}$

EX: The CDG G_{abc} :

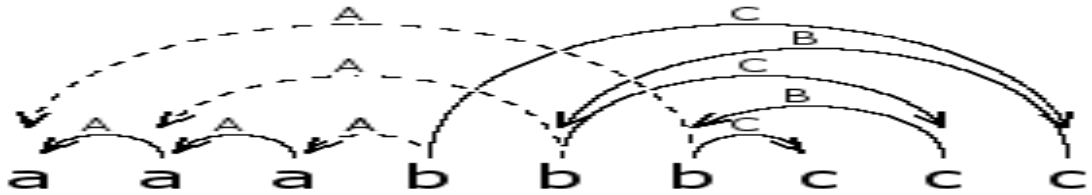
$a \mapsto A^{\swarrow A},\ [A\backslash A]^{\swarrow A},$

$b \mapsto [B/C]^{\nwarrow A},\ [A\backslash S/C]^{\nwarrow A},$

$c \mapsto C,\ [B\backslash C]$

generates the language $\{a^n b^n c^n \mid n > 0\}$.

E.g., $a^3 b^3 c^3$ obtains the DS:



due to the proof:

$$\dfrac{\dfrac{\dfrac{[A]^{\swarrow A}\quad [A\backslash A]^{\swarrow A}}{[A]^{\swarrow A\swarrow A}}(\mathbf{L}^l)\quad [A\backslash A]^{\swarrow A}}{[A]^{\swarrow A\swarrow A\swarrow A}}(\mathbf{L}^l)\quad \dfrac{\dfrac{[B/C]^{\nwarrow A}\quad \dfrac{\dfrac{[B/C]^{\nwarrow A}C}{B^{\nwarrow A}}(\mathbf{L}^r)\quad [B\backslash C]}{C^{\nwarrow A}}(\mathbf{L}^l)}{B^{\nwarrow A\nwarrow A}}(\mathbf{L}^r)\quad [B\backslash C]}{C^{\nwarrow A\nwarrow A}}(\mathbf{L}^l)\quad [A\backslash S/C]^{\nwarrow A}}{[A\backslash S]^{\nwarrow A\nwarrow A\nwarrow A}}(\mathbf{L}^l)}{[S]^{\swarrow A\swarrow A\swarrow A\nwarrow A\nwarrow A\nwarrow A}}(\mathbf{L}^l)}{S}(\mathbf{D}^l\times 3)$$

CDG of French

Parameters of CDG of French constructed by bootstrapping from examples:

Examples	Classes			Regular Expressions	Dependencies	
	total	verbal	nominal		projective	distant
500	190	46	6	~ 2500	$85(9\ par)$	$22(3\ par)$

(where $n(m\ par)$ means n of which m are parametrized)

The grammar is developed by A. Dikovsky during 9 months (beginning of June 2009 - end of March 2010).

It explains :

- the kernel syntax (copulas/auxiliary verbs, verb complements (a-obj, d-obj, g-obj, l-obj, o-obj, Agent), noun modifiers, verb circumstantials, pronouns),
- genitive/partitive,
- negation (*ne..pas* (etc.), *ne..que*),
- clitics and reflexives,
- light verbs and infinitives,
- comparative constructions,
- clauses, coordination,
- word order inversion : interrogative sentences, topicalized complements,
- vocative constructions, co-reference, expletives (les incises),
- aggregation,
- complex lexical units and numbers, unknown terms.

Integration of a morphological dictionary

PROBLEM : Lexicon of the bootstrapped CDG $G_{appx}[T]$ is limited to the examples from which it was constructed (~ 2000 words).

To extend it to a wide scope French lexicon, was used the open resource morphological dictionary of French Lefff 3.0 [B.Sagot'08].

Lefff 3.0 has more than 500000 forms annotated with morpho-syntactic feature values:

EX : $octroierions\ 100\ v\ [pred =' octroyer\_\_\_\_\_1$
$< Suj : cln|scompl|sinf|sn, Obj : (cla|sn) >$,
$@pers, cat = v,\ @C1p]\ octroyer\_\_\_\_\_1\ Default\ C1p$

Integration of Lefff with CDG : $G_{appx}[T]$:

- $G_{appx}[T]$ and Lefff are stored in the RDB *PostgreSQL* (R.Alfared)
- correspondence between CDG classes and the corresponding Lefff forms is realized through SQL queries or manually (R.Alfared, D.Béchet, A.Dikovsky, during 5 months).