Generic Axiomatization of Families of Noncrossing Graphs in Dependency Parsing

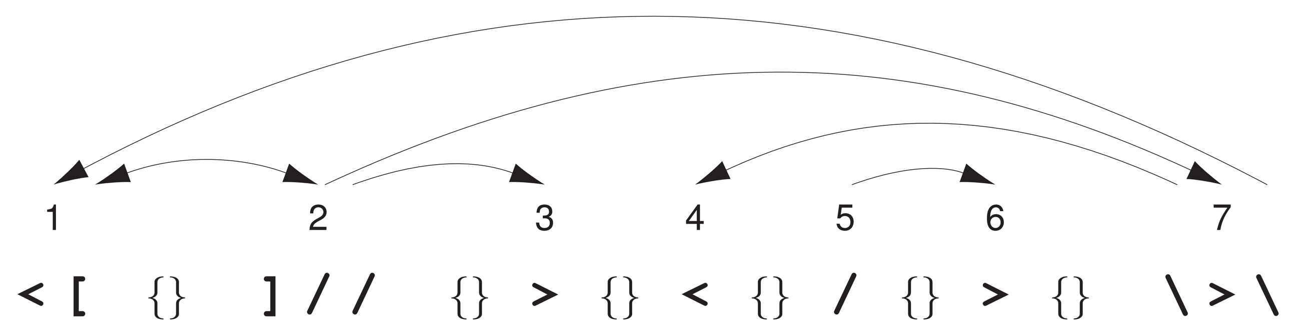
Anssi Yli-Jyrä (University of Helsinki) and Carlos Gómez-Rodríguez (Universidade da Coruña)

Abstract

1. We develop a simple linear encoding supporting general noncrossing digraphs.
2. We show that the encoded noncrossing digraphs form a context-free language.
3. We present an latent encoding that can be used to characterize various families of digraphs by forbidden local patterns.
4. This can be used to enable generic context-free parsers that produce different families of noncrossing graphs with the same set of inference rules.

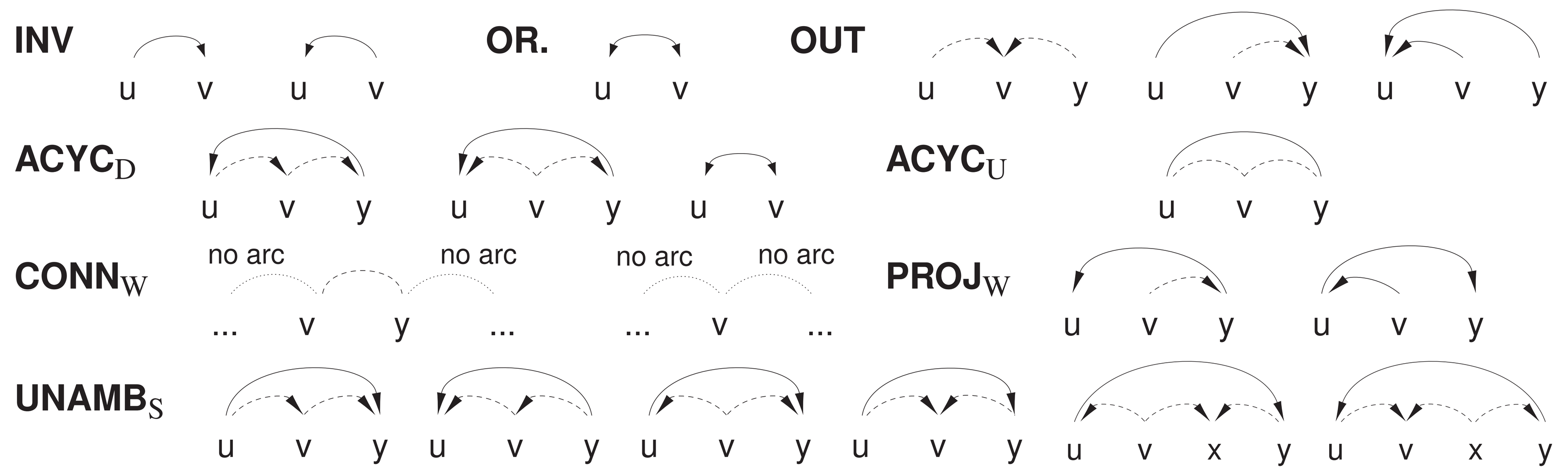
Noncrossing Digraphs as Code Strings

Enc : **NC-DIGRAPH** $\leftrightarrow$ $L_{\textbf{NC-DIGRAPH}}$



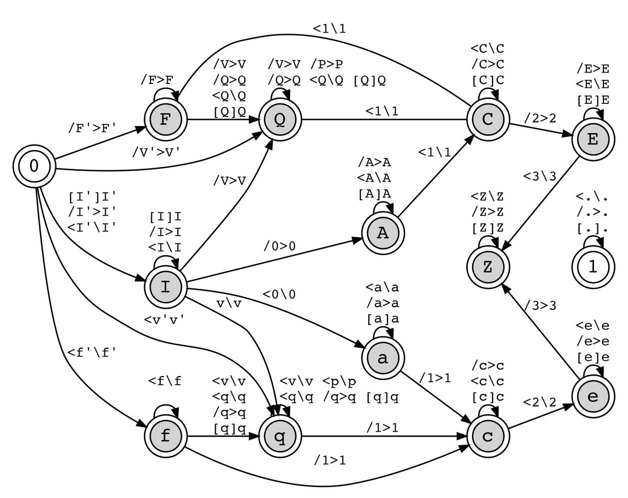
Axioms and Forbidden Patterns in Noncrossing Digraphs

Bounded treewidth $\Rightarrow$ MSO properties become LOGSPACE decidable (by **Courcelle's theorem**)

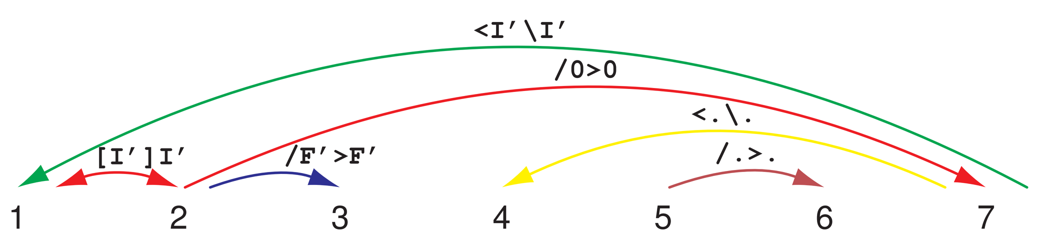


Latent Edge Types

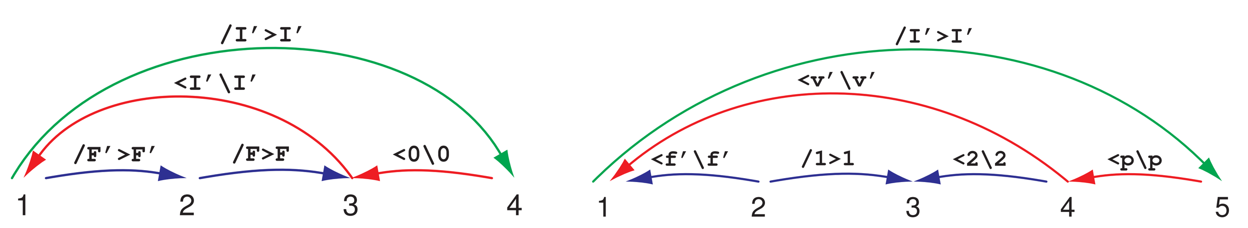
A **chain** consists of contiguous linear edge brackets, e.g. [[] [] []] [] (2-edge-chain; 3-edge chain)
A **loose chain** starts immediately after a word boundary {}.
A **local automaton** *Chains* decorates the chains with **latent edge types.**



′	from the initial state 0 for a non-loose chain;
.	from the initial (and only) state 1 for a loose chain;
I	a bidirectional chain: $u \leftrightarrow (v \leftrightarrow)y$;
0A	a primarily bidirectional forward chain: $u \leftrightarrow v \rightarrow y$;
F	a forward chain: $u \rightarrow v \rightarrow y$;
VQP	a primarily forward chain: $u \rightarrow v \leftrightarrow (\cdots \rightarrow)y$;
1C	a primarily forward 1-turn chain: $u \rightarrow v \leftarrow y$;
2E	a primarily forward 2-turn chain: $u \rightarrow v \leftarrow x \rightarrow y$;
3Z	a 3-turn chain;
0afvqp1c2e	(analogously for primarily backward chains)

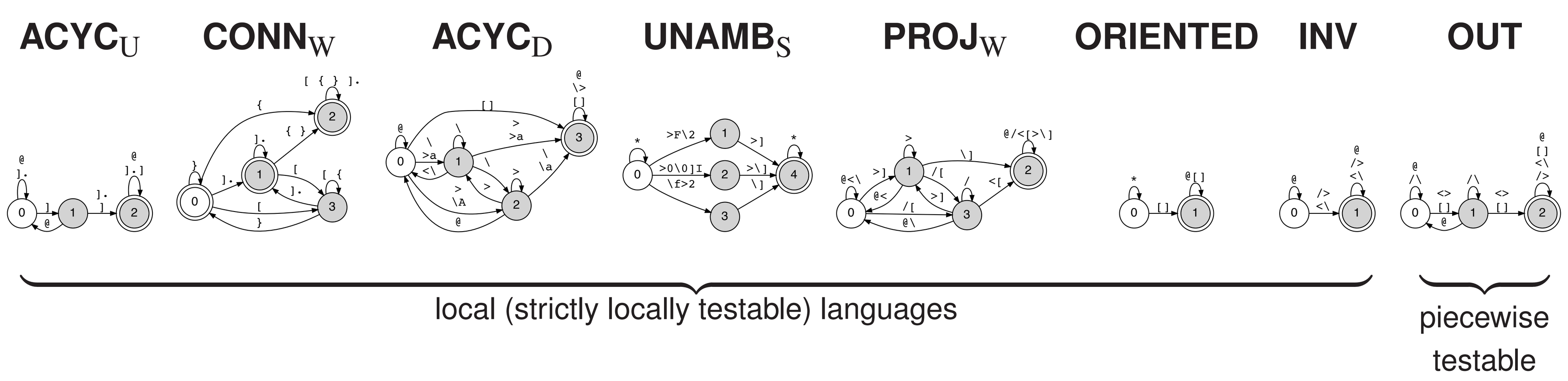


Local automata *Looseness* and *Covered* select the initial state and some further edge subtypes in *Chains*, respectively. The subtypes of edges are defined according to the chains they cover. The bracket types I, Q, q and the brackets \A, >a, >C, \c, \E, >e indicate arcs that constitute a cycle with the chain they cover and the bracket types V and v indicate arcs that cover 2-turn chains:



Deciding Forbidden Patterns in Digraphs via Star-Free Finite State Constraints

The forbidden patterns become **star-free (FO local)** and decidable in **deterministic linear time**.

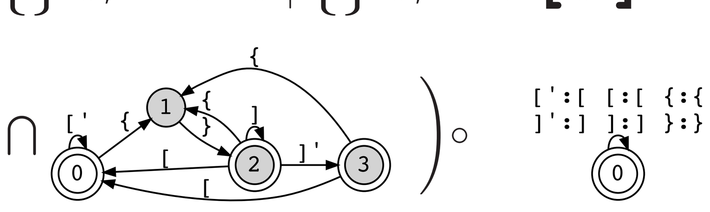


Three Representations for $L_{\textbf{NC-DIGRAPH}}$

$L_{\textbf{NC-GRAPH}}$ is an unambiguous CFL and a subset of D_2, a **Dyck** language over letters [,], {, }.

derivational representation: $S \rightarrow BS \mid \{\}\, S \mid \epsilon;\ S' \rightarrow BT \mid \{\}\, S;\ T \rightarrow BS \mid \{\}\, S;\ B \rightarrow [\, S' \,]$

1st morphic representation: $(D_3 \cap Reg) \circ h = \left(\begin{array}{l} S \rightarrow [S]\ S \\ S \rightarrow [' S]'\ S \\ S \rightarrow \{S\}\ S \\ S \rightarrow \epsilon \end{array} \cap \right) \circ$



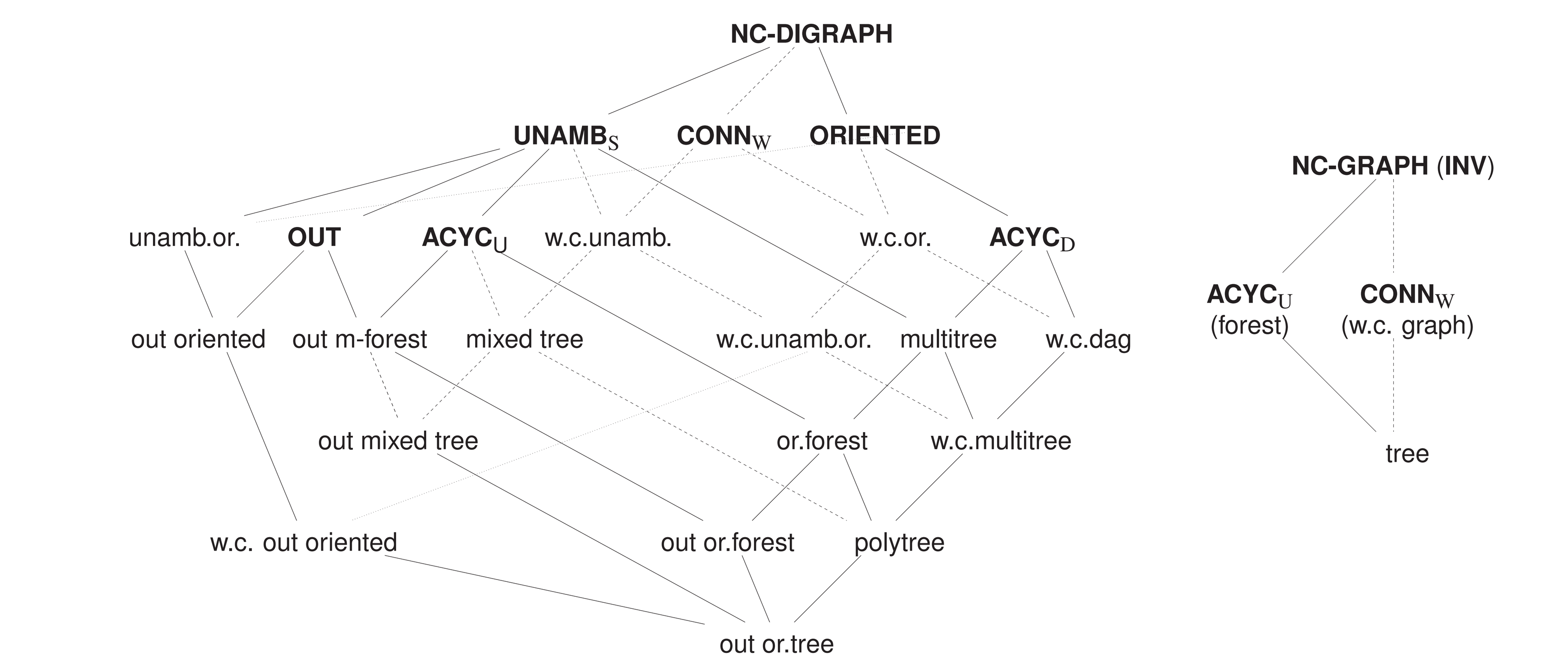
There are similar representations for $L_{\textbf{NC-DIGRAPH}} \subset D_4$ having letters [,], /, >, <, \, {, }.
Latent edge types are distinguished in the **internal language**

$$L_{\textbf{NC-DIGRAPH}_{\text{lat}}} = D_{55} \cap (Reg' \cap Chains \cap Looseness \cap Covered)$$

This give rise to the third representation for $L_{\textbf{NC-DIGRAPH}}$:

2nd morphic representation: $(D_{55} \cap Reg_{\text{lat}}) \circ h_{\text{lat}} = L_{\textbf{NC-DIGRAPH}_{\text{lat}}} \circ h_{\text{lat}}$

Generic Representation for the Subfamilies of Digraphs



All elements of the $L_{\textbf{NC-DIGRAPH}}$ ontology are **unambiguous** and **closed under intersection**.

Enumeration Experiment per n Nodes

Name	Sequence prefix for $n = 2, 3, \ldots$	Example	Name	Sequence prefix for $n = 2, 3, \ldots$	Example
digraph	**(KJ)**: 4,64,1792,**62464**,2437120,101859328 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat})$	1 2 3 4 5	weakly projective digraph	4,36,480,**7744**,138880,2661376 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap P_W)$	1 2 3 4 5
w.c.digraph	3,54,1539,**53298**,2051406,84339468 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap C_W)$	1 2 3 4 5	w.p. w.c.digraph	3,26,339,**5278**,90686,1658772 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap P_W \cap C_W)$	1 2 3 4 5
unamb.digr.	4,39,529,**8333**,142995,2594378 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap U_S)$	1 2 3 4 5	w.p. unamb.digr.	4,29,275,**3008**,35884,453489 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap P_W \cap U_S)$	1 2 3 4 5
m-forest	4,37,469,**6871**,109369,1837396,32062711 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap A_U)$	1 2 3 4 5	w.p. m-forest	4,29,273,**2939**,34273,421336 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap P_W \cap A_U)$	1 2 3 4 5
out digraph	4,27,207,**1683**,14229,123840,1102365 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap Out)$	1 2 3 4 5	w.p. out digraph	4,21,129,**867**,6177,45840,350379 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap P_W \cap Out)$	1 2 3 4 5
or. digraph	3,27,405,**7533**,156735,3492639,77539113 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap O)$	1 2 3 4 5	w.p. or.digraph	see w.p.dag $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap P_W \cap O)$	see w.p.dag
dags	**(A246756):** 3,25,335,**5521**,101551 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap A_D)$	1 2 3 4 5	w.p. dag	3,21,219,**2757**,38523, 574725, 8967675 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap P_W \cap A_D)$	1 2 3 4 5
w.c. dag	**(KJ)**: 2,18,242,**3890**,69074,1306466 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap A_D \cap C_W)$	1 2 3 4 5	w.p. w.c. dag	2,14,142,**1706**,22554,316998,4480592 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap P_W \cap A_D \cap C_W)$	1 2 3 4 5
multitree	3,19,167,**1721**,19447,233283,2917843 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap A_D \cap U_S)$	see oriented forest or w.c. multitree	w.p. multitree	3,17,129,**1139**,11005,112797,1203595 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap P_W \cap A_D \cap U_S)$	1 2 3 4 5
or.forest	3,19,165,**1661**,18191,210407,2528777 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap A_D \cup A_U)$	1 2 3 4 5	w.p. or.forest	3,17,127,**1089**,10127,99329,1010189 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap P_W \cap A_D \cup A_U)$	1 2 3 4 5
w.c. multitree	2,12,98,**930**,9638,105798,1201062 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap A_D \cap U_S \cap C_W)$	1 2 3 4 5	w.p. w.c. multitree	2,10,68,**538**,4650,42572,404354 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap P_W \cap A_D \cap U_S \cap C_W)$	1 2 3 4 5
out or.forest	3,16,105,**756**,5738,45088,363221 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap A_D \cap Out)$	1 2 3 4 5	w.p. out or.forest	**(A003169):** 3,14,79,**494**,3294,22952 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap P_W \cap A_D \cap Out)$	1 2 3 4 5
polytree	**(A153231):** 2,12,96,**880**,8736,91392 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap A_D \cap C_W \cap A_U)$	1 2 3 4 5	w.p. polytree	**(A027307):**2,10,66,**498**,4066,34970 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap P_W \cap A_D \cap C_W \cap A_U)$	1 2 3 4 5
out or.tree	**(A174687):** 2,9,48,**275**,1638,9996 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap A_D \cap C_W \cap Out)$	1 2 3 4 5	projective out or.tree	**(A006013):** 2,7,30,**143**,728,3876,21318 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap P_W \cap A_D \cap C_W \cap Out)$	1 2 3 4 5
graph	**(A054726):** 2,8,48,**352**,2880,25216 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap I)$	1 2 3 4 5	connected graph	**(A007297):** 1,4,23,**156**,1162,9192 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap I \cap C_W)$	1 2 3 4 5
forest	**(A054727):** 2,7,33,**181**,1083,6854 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap I \cap A_U)$	1 2 3 4 5	tree	**(A001764,YJ):** 1,3,12,**55**,273,1428,7752 $h_{lat}(D_{55} \cap G_n \cap Reg_{lat} \cap I \cap A_U \cap C_W)$	1 2 3 4 5

The correctness was verified against OEIS, the prior art, and procedural enumerate-test algorithms.

Application to Generic Parsing

- ***n*-Node Digraphs:** $L_{\textbf{NC-DIGRAPH}} \cap G_n$ where $G_n = \overline{B}^*(\{\}\overline{B}^*)^{n-1}$ and $B = \{$curly brackets$\}$.
- **Arc-Factored Parsing:** Each possible arc (i, j) has a positive weight defined e.g. by $w_{ij} = \mathbf{w} \cdot \Phi(sentence, (i, j))$. The parsing maximises the total weight of arcs:

$$A = \arg\max_{A \in L_{\text{family of NC-DIGRAPHs}} \cap G_n} \sum_{(i,j) \in A} w_{i,j}$$

- **Indexed brackets:** Edges in $[_1[_1\{\}]_2[_2\{\}\{\}]_4]_4$ get weights from a Dyck grammar:

$$S \rightarrow \epsilon \mid \{\};\ S \overset{w_{12}}{\rightarrow} [_1S]_2S;\ S \overset{w_{13}}{\rightarrow} [_1S]_3S;\ S \overset{w_{14}}{\rightarrow} [_1S]_4S;$$
$$S \overset{w_{23}}{\rightarrow} [_2S]_3S;\ S \overset{w_{24}}{\rightarrow} [_2S]_4S;\ S \overset{w_{34}}{\rightarrow} [_3S]_4S.$$

- **Intersection:** $(D_{55} \cap Reg_{lat} \cap G_n \cap Constraints)$ gives a weighted CFG.
- **Dynamic Programming:** The arg max inference reduces to WCFG parsing.
- **Lexicalized Search Space:** The axioms and lexical contraints on the feasible brackets for each token can be implemented in lexical entries (compare: multi-modal CCG) that refine G_n.

Conclusion: Four Contributions

1. **Linear Encoding (*Enc*):**
   Noncrossing digraphs encoded bijectively as strings that constitute a context-free subset of D_4.
2. **Context-Free Axioms:**
   The current MSO definable axioms become unambiguous CF languages
   - axioms become star-free (mostly local) constraints for latent bracketing
   - cf. linear time and LOGSPACE testability of MSO under bounded treewidth (Courcelle 1990; Elberfeld et al. 2010)
3. **Ontology of Digraphs:**
   The axioms generate a semi-lattice containing 12 known categories plus many new ones.
4. **Generic parsing:**
   One parser or enumeration algorithm for all families of noncrossing digraphs.
   - Weighted CF parsing with dynamic programming
   - Inference with constraint relaxation
   - Lexical control over digraph properties

Contact Information

anssi.yli-jyra@helsinki.fi
Department of Modern Languages
University of Helsinki

carlos.gomez@udc.es
Departamento de Computación
Universidade da Coruña

Acknowledgements

- The Academy of Finland (dec. No 270354 - A Usable Finite-State Model for Adequate Syntactic Complexity)
- The University of Helsinki (dec. RP 137/2013 - SMT based on D-Tree Contraction Grammars and FSTs)
- The European Research Council (ERC): Horizon 2020 (agr. No 714150 - FASTPARSE)
- TELEPARES-UDC project (FFI2014-51978-C2-2-R) from MINECO.