CLAMR: Cross-Lingual Abstract Meaning Representations for Machine Translation

Ondřej Bojar, Wei-Te Chen, Silvie Cinkova, David Chiang, Frank Drewes, Ondřej Dušek, Jeff Flanigan, Dan Gildea, Jan Hajič, Alexander Koller, Adam Lopez, Tim O'Gorman, Martha Palmer, Xiaochang Peng, Martin Popel, Adithya Renduchintala, Naomi Saphra, Giorgio Satta, Roman Sudarikov, Zdeňka Urešová, Chuan Wang, Nianwen Xue, Yuchen Zhang

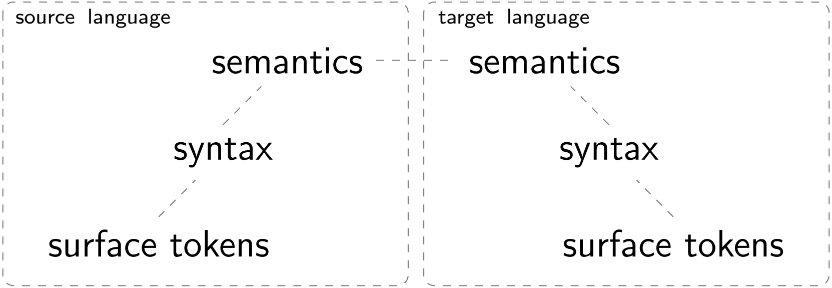
Warren Weaver's letter (1949)

Think, by analogy, of individuals living in a series of tall closed towers, all erected over a common foundation. When they try to communicate with one another they shout back and forth, each from his own closed tower. It is difficult to make the sound penetrate even the nearest tower, and communication proceeds very poorly indeed. But when an individual goes down his tower, he finds himself in a great open basement, common to all the towers. Here he establishes easy and useful communication with the persons who have also descended from their towers.

Warren Weaver's letter (1949)

Thus may it be true that the way to translate from Chinese to Arabic, or from Russian to Portuguese, is not to attempt the direct route, shouting from tower to tower. Perhaps the way is to descend, from each language, down to the common base of human communication - the real but as yet undiscovered universal language - and then re-emerge by whatever particular route is convenient.

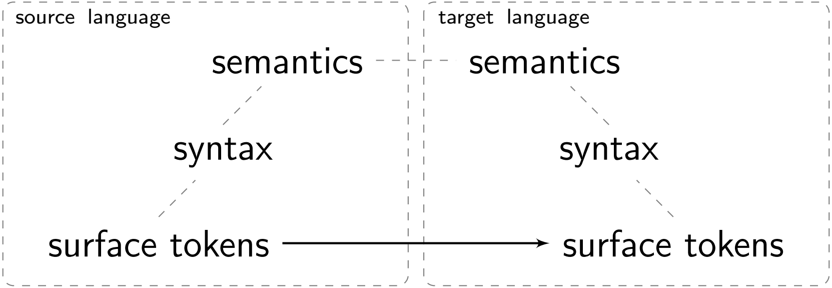
The Vauquois triangle



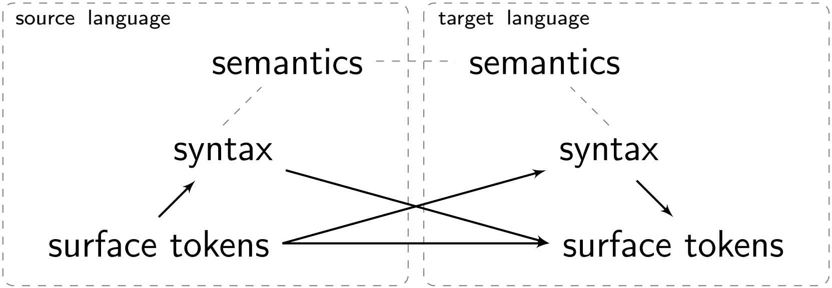
Warren Weaver's letter (1949)

One naturally wonders if the problem of translation could conceivably be treated as a problem in cryptography. When I look at an article in Russian, I say This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.

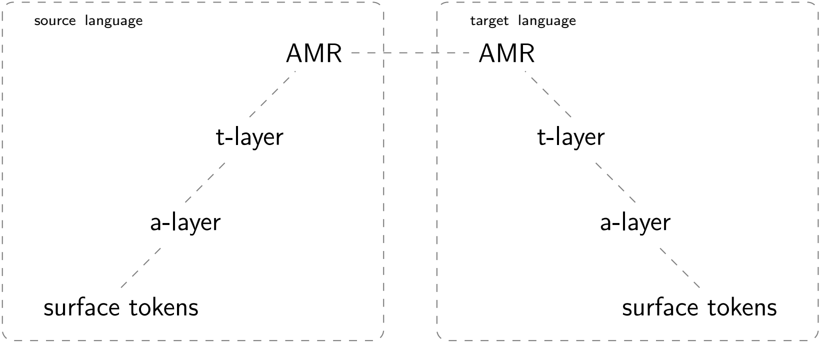
The IBM approach



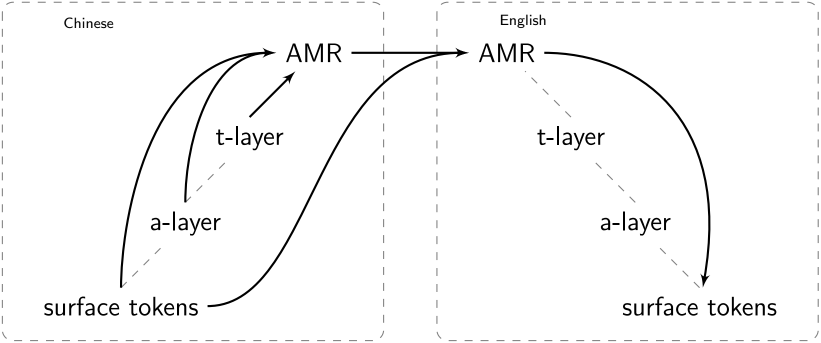
The last decade



Last Month: Cross-Lingual AMR data



Today: Cross-Lingual AMR systems



Warren Weaver's letter (1949)

It is one of the chief purposes of this memorandum to emphasize that statistical semantic studies should be undertaken, as a necessary preliminary step.

Part I

GLAMR: Graph Languages for AMR

David Chiang, Frank Drewes, Daniel Gildea, Alexander Koller, Adam Lopez, Naomi Saphra, Giorgio Satta

Example

input	gimme suttin 2 beleive innnn
output	give me something to believe in

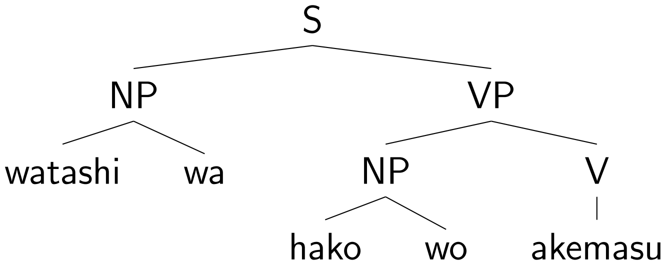
Example

input	ostoskeskuksessa
output	ostos#keskus+N+Sg+Ine

Example

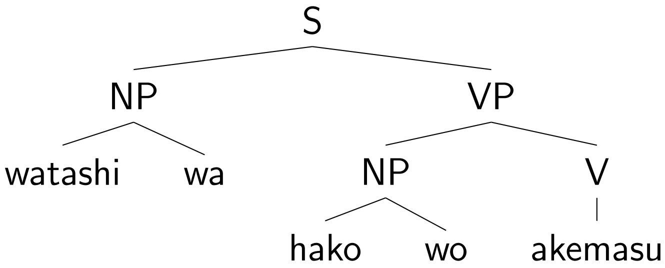
input watashi wa hako wo akemasu.

output

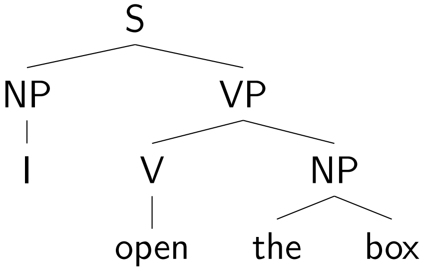


Example

input



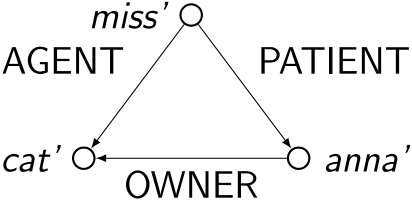
output



Example

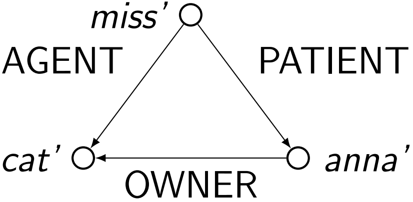
input Anna fehlt ihrem Kater

output



Example

input



output Anna's cat is missing her

Common Problems

1. Inputs/outputs are strings, trees, and graphs
2. Must cope with ambiguity

Common Solutions

1. Automata and transducers
2. Probability

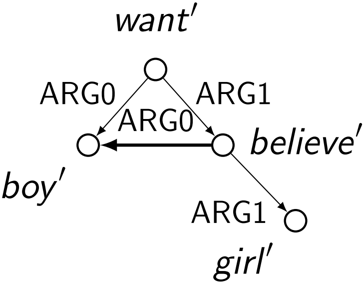
Starting Point

	Strings	Trees	Graphs
acceptors	yes	yes	
transducers	yes	yes	
probabilistic	yes	yes	
recognition	yes	yes	
intersection	yes	yes	
summation	yes	yes	
implementation	yes*	yes†	

*OpenFST (Google), used by Kaldi ASR, Cambridge MT

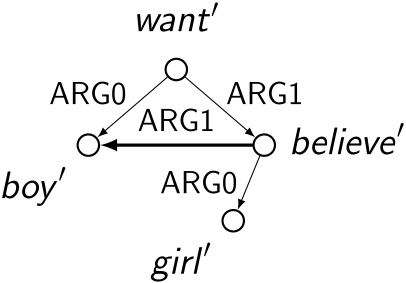
†Tiburon (ISI)

Trees vs. Graphs: Reentrancies



The boy wants **[PRO]** to believe the girl.

local



The boy wants the girl to believe **him**.

nonlocal

Starting Point: Graph Grammars

	RTGs	DAGAs*	HRGs
reentrancies			
local	**no**	yes?	yes
nonlocal	**no**	yes	**no**
acceptors	yes	yes	yes
transducers	yes	yes†	yes
probabilistic	yes	**no**	yes
recognition $\mathcal{O}(\cdot)$	$q^{d+1}n$	2^n	$(3^d n)^{k+1}$
intersection	yes	yes	yes‡
summation	yes	?	yes

*Nonplanar variant of Quernheim and Knight (2012)

†To trees

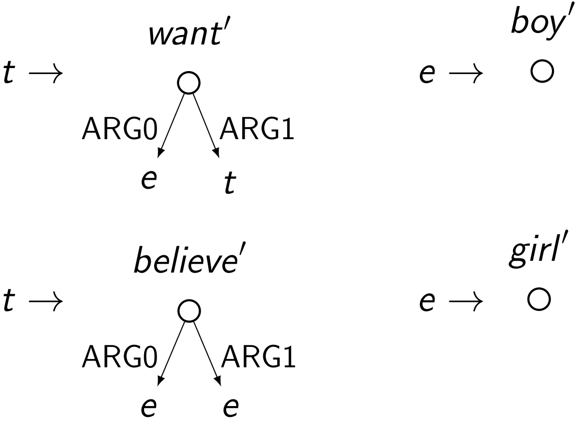
‡Of derivations, not graphs

Section 2

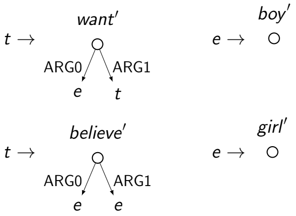
Regular graph grammars

Regular graph grammars

Simplified version of nonplanar DAG acceptors (Quernheim and Knight, 2010)



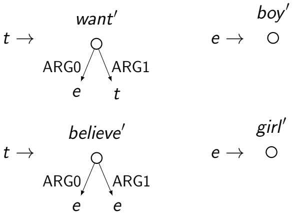
Regular graph grammars

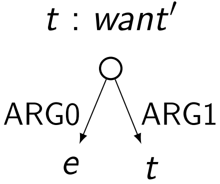


t

The boy wants the girl to believe him.

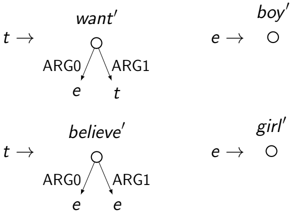
Regular graph grammars

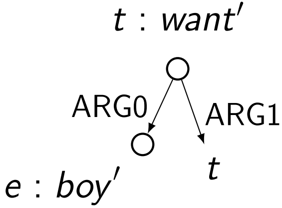




The boy wants the girl to believe him.

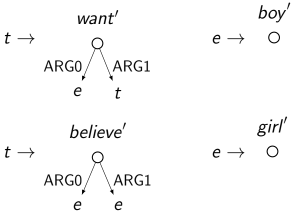
Regular graph grammars

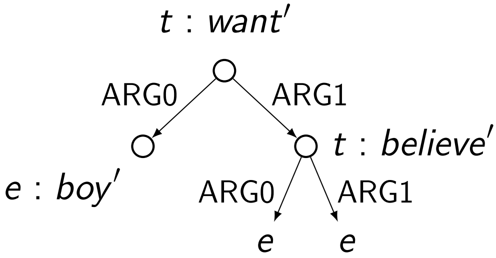




The boy wants the girl to believe him.

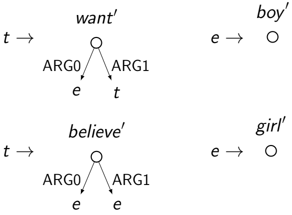
Regular graph grammars

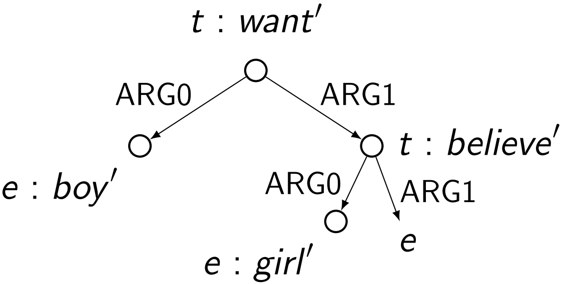




The boy wants the girl to believe him.

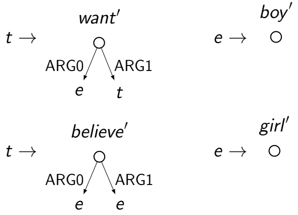
Regular graph grammars

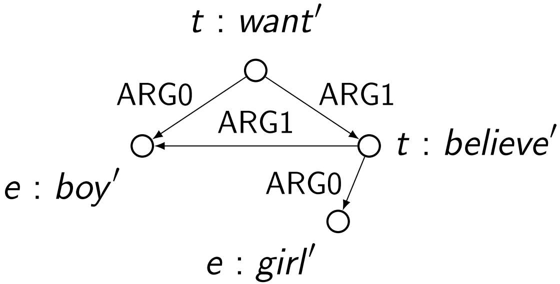




The boy wants the girl to believe him.

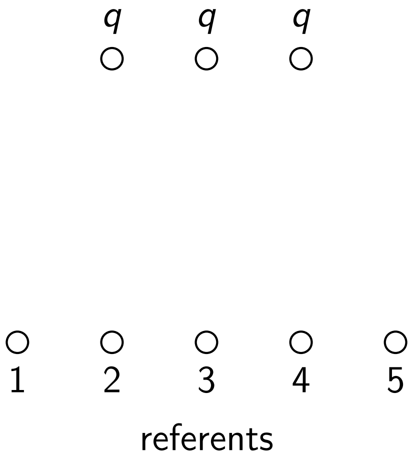
Regular graph grammars



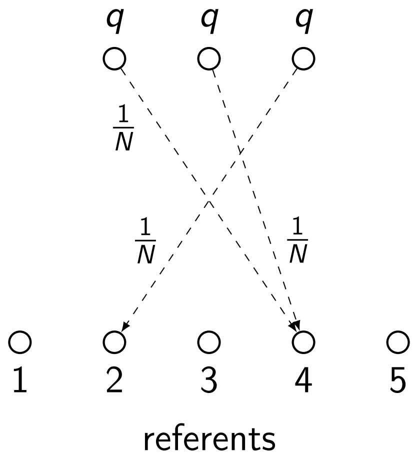


The boy wants the girl to believe him.

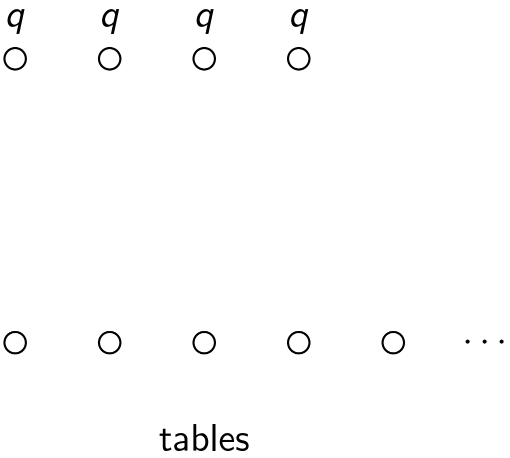
Probabilities



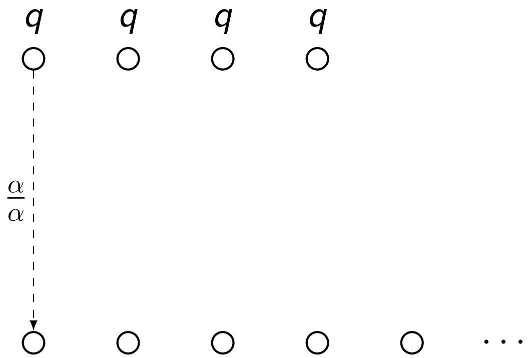
Probabilities



Probabilities: Chinese restaurant process

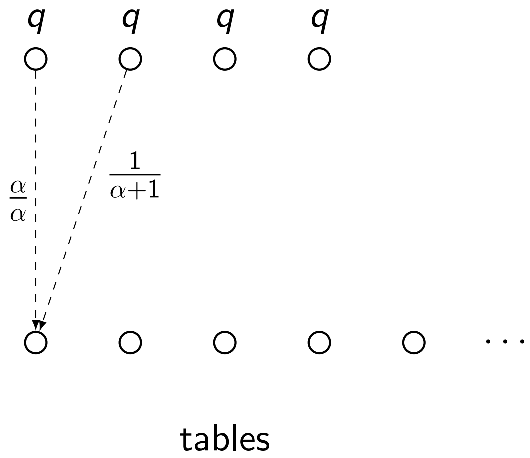


Probabilities: Chinese restaurant process

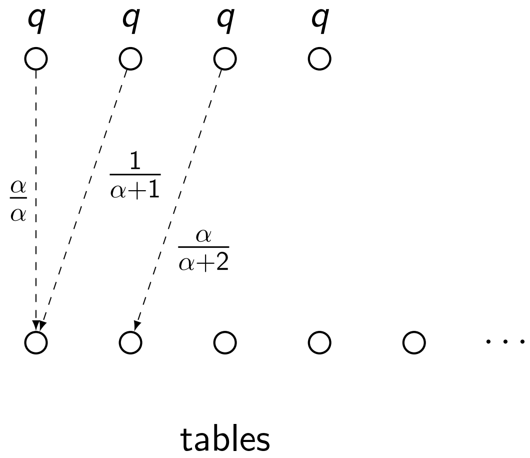


tables

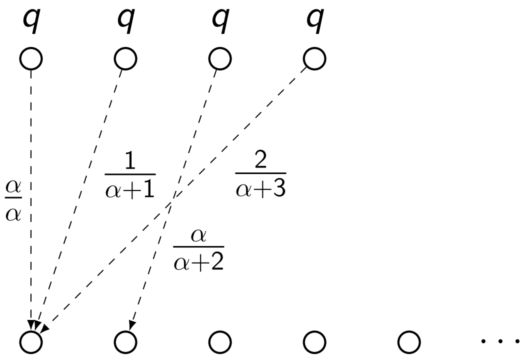
Probabilities: Chinese restaurant process



Probabilities: Chinese restaurant process



Probabilities: Chinese restaurant process

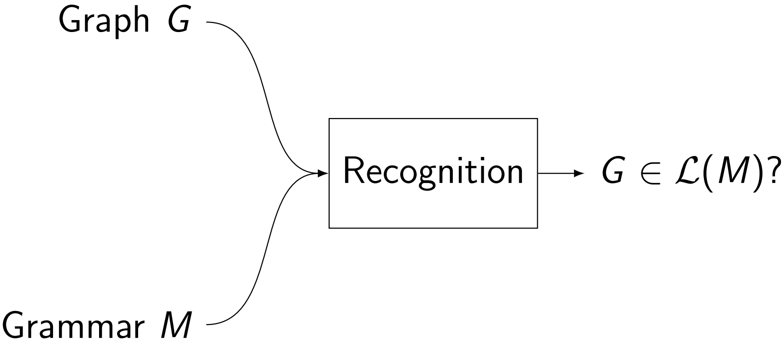


tables

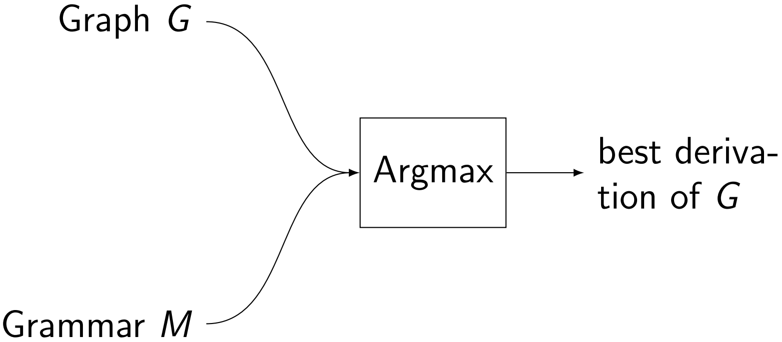
Probabilities

	referents	CRP
maximum nodes	N	∞
hyperparameters	N	α
equivalent mergings	distinct	summed
independent	yes	no
exchangeable	yes	yes

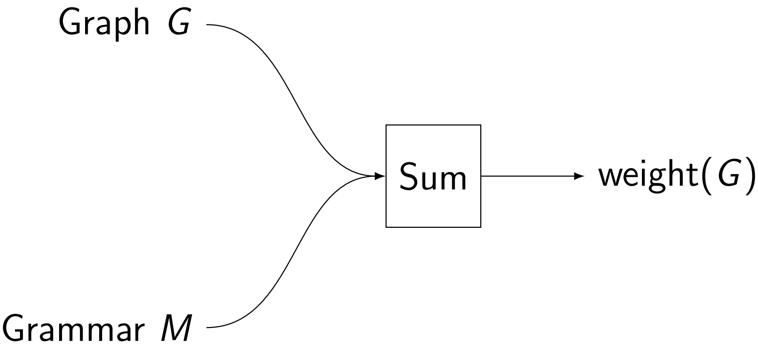
Recognition



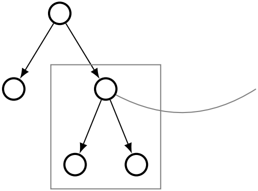
Max-weight derivation



Total weight



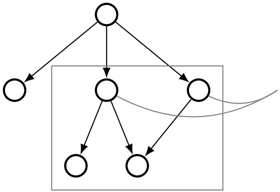
Recognition: Algorithm



$$\left\{ \begin{array}{l} q_1 \mapsto 0 \\ q_2 \mapsto 1 \\ q_3 \mapsto 0 \end{array} \right\}$$

Value for subtree depends only on state at root

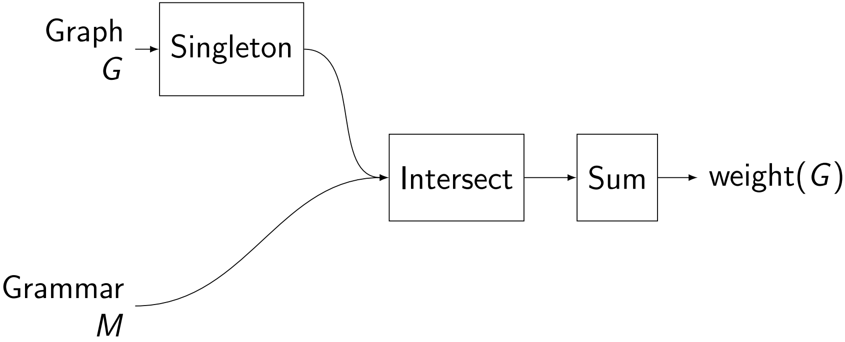
Recognition: Algorithm



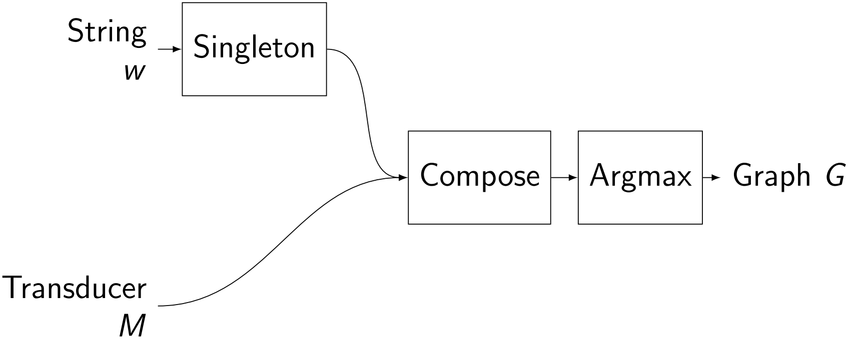
$$\left\{ \begin{array}{l} q_1, q_4 \mapsto 0 \\ q_1, q_5 \mapsto 1 \\ q_2, q_4 \mapsto 0 \\ q_2, q_5 \mapsto 1 \\ q_3, q_4 \mapsto 1 \\ q_3, q_5 \mapsto 1 \end{array} \right\}$$

- Time complexity: $\mathcal{O}(q^{treewidth+1} n)$
- Average treewidth: 1.55 *but* need to add edges between sisters *but* Kleene star will help
- Implemented in Julia (by Naomi Saphra)

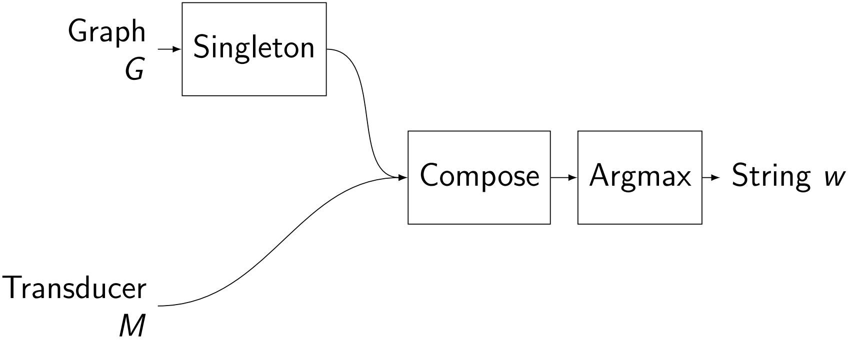
Recognition as intersection



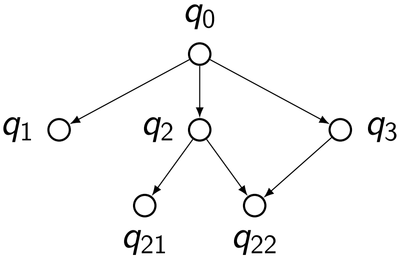
Interpretation as composition

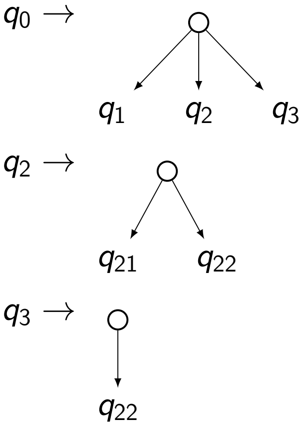


Generation as composition



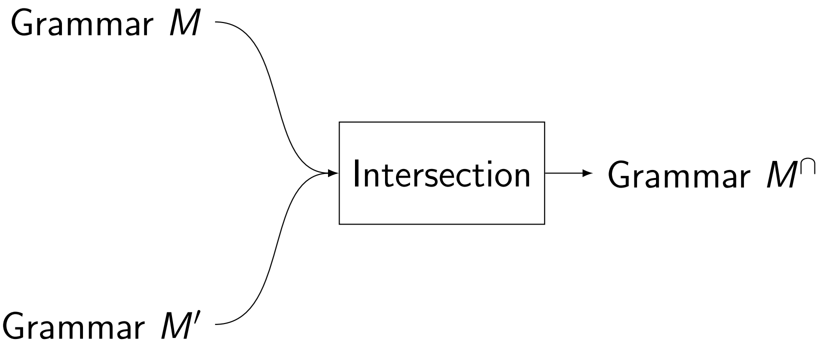
Grammar for a single graph





Unique(q_{22})

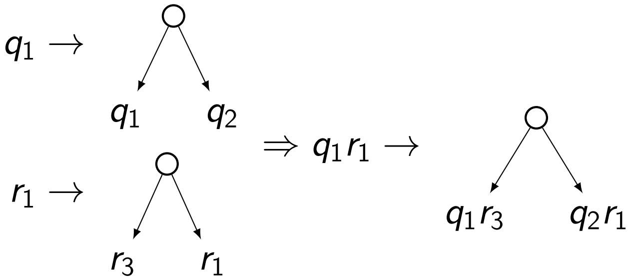
Intersection



$$\mathcal{L}(M^{\cap}) = \mathcal{L}(M) \cap \mathcal{L}(M')$$

Intersection: Algorithm

	r_1	r_2	r_3
q_1	$q_1 r_1$	$q_1 r_2$	$q_1 r_3$
q_2	$q_2 r_1$	$q_2 r_2$	$q_2 r_3$



Intersection: Algorithm

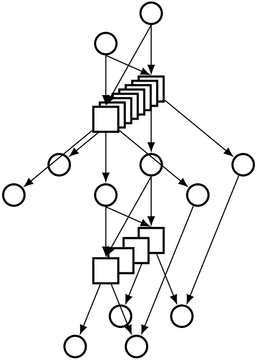
$$\begin{array}{c|ccc} & r_1 & r_2 & r_3 \\ \hline q_1 & q_1 r_1 & q_1 r_2 & q_1 r_3 \\ q_2 & q_2 r_1 & q_2 r_2 & q_2 r_3 \end{array}$$

$$\text{Unique}(q_2) \Rightarrow \text{Unique}(q_2 r_1, q_2 r_2, q_2 r_3)$$

Only one node in *any* of the states can exist (other states cannot be used)

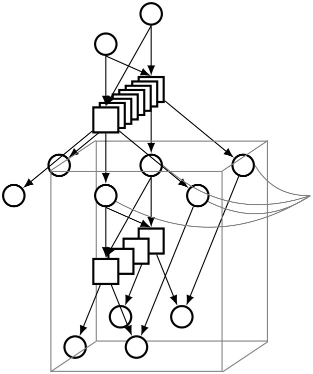
Hypergraphs

Grammar for a finite language can be represented as a hypergraph (analogous to lattices, packed forests).



Note: not all edges are drawn

Summation



$$\left\{ \begin{array}{c} q_2 r_1, q_3 r_1 \mapsto 0.2 \\ q_2 r_2, q_3 r_1 \mapsto 0.3 \\ q_3 r_1, q_3 r_2 \mapsto 0.5 \\ q_2 r_2, q_3 r_2 \mapsto 0.4 \\ \vdots \end{array} \right\}$$

- Time complexity: $\mathcal{O}(q^{treewidth+1} n)$
- Byproduct: hyperedge replacement grammar

Summary

	RTGs	DAGAs	HRGs	RGGs
reentrancies				
local	**no**	yes?	yes	yes?
nonlocal	**no**	yes	**no**	yes
acceptors	yes	yes	yes	yes
transducers	yes	yes	yes	yes?
probabilistic	yes	**no**	yes	yes*
recognition $\mathcal{O}(\cdot)$	$q^{d+1}n$	2^n	$(3^d n)^{k+1}$	$q^{k+1}n$
intersection	yes	yes	yes	yes
summation	yes	?	yes	yes

*But nicest model makes algorithms harder

Section 3

Follow-Up Projects

Eliminate 3^d for HRG

- General algorithm for parsing graphs with HRG is $O((3^d n)^{k+1})$
  - n: size of graph
  - k: treewidth
  - d: degree of graph
- Known conditions for polynomial-time parsing:
  - separability: removing s nodes breaks graph into $O(\log n)$ components
  - componentwise derivation: remove node of high degree, derive components independently
- AMRs: nodes of high degree are often implicit arguments (first condition) or conjunctions (second condition)

Bag of Node Parsing

- Words observed in sentence $\Rightarrow$ nodes in graph
- Node parsing problem: find all graphs generated by grammar $\mathcal{G}$ having bag of nodes V

Synchronous RGG

- Combining RGG derivation with CFG (or dependency) derivation
- Node merge operation: pronouns?
- Are derivation trees on RGG side projective on string side?
  - Mildly context sensitive on string side?

HRGs with Node Merge

- Syntactic re-entrancies (control verbs) generated by HRG
- Non-local re-entrancies (pronouns) generated by RGG merge operation