Evaluating grammar formalisms for applications

to natural language processing and biological sequence analysis

David Chiang

28 June 2004

Applications of grammars

- Statistical parsing (Charniak, 1997; Collins, 1997)
- Language modeling (Chelba and Jelinek, 1998)
- Statistical machine translation (Wu, 1997; Yamada and Knight, 2001)
- Prediction or modeling of RNA/protein structure (Searls, 1992)

Applications of grammars

- Grammars are a convenient way to. . .
    - encode bits of theories (subcategorization, SVO/SOV/VSO)
    - structure algorithms (searching through word alignments, chain foldings)
- A difficulty of using grammars: don’t know what kind to use

The overarching question

What makes one grammar better than another?

- Weak generative capacity (WGC): what *strings* does a grammar generate?
- Strong generative capacity (SGC): what *structural descriptions* (SDs) does a grammar generate?
  - specifies whatever is needed to determine how the sentence is used and understood (Chomsky)
  - not just phrase-structure trees

Weak vs. strong generative capacity

- Chomsky:
  - WGC is “the only area in which substantial results of a mathematical character have been achieved”
  - SGC is “by far the more interesting notion”
- Theory focuses on WGC because it’s easier to compare strings than to compare SDs
- Applications are concerned with SGC because SDs contain the information that eventually gets used
- Occasional treatment of SGC (Kuroda, 1976; Miller 1999) but nothing directed towards computational applications

Objective

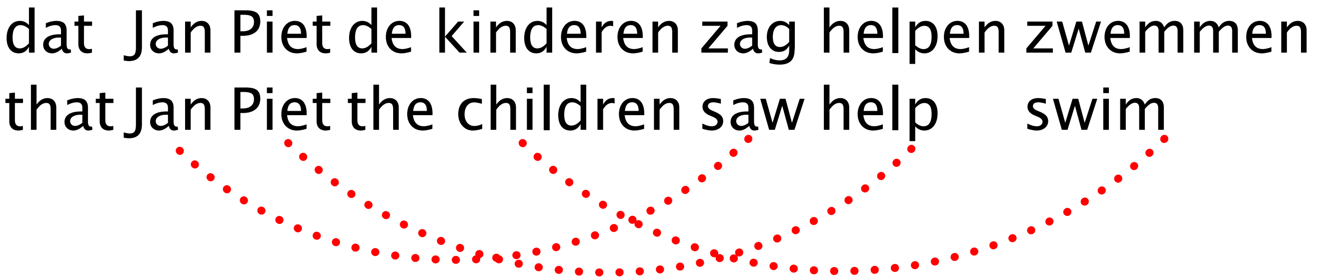
- Ask the right questions: refine SGC so that it is rigorous (unlike before) and relevant (unlike WGC) to applications

- Answer the questions and see what the consequences are for applications

- Three areas:
  - Statistical natural language parsing
  - Natural language translation
  - Biological sequence analysis

Historical example: cross-serial dependencies

- Example from Dutch:



dat Jan Piet de kinderen zag helpen zwemmen
that Jan Piet the children saw help swim

‘that Jan saw Piet help the children swim’

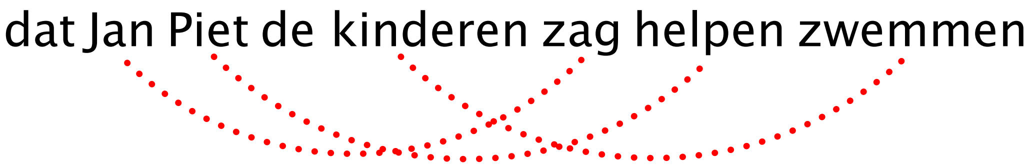
- Looks like non-context-free $\{ww\}$ but actually context-free, like $\{\mathrm{a}^n\mathrm{b}^n\}$ (Pullum and Gazdar, 1982)

- How to express intuition that this is beyond the power of CFG?

Historical example: a solution

Two things had to happen to show this was beyond CFG but within TAG (Joshi, 1985):

1. A different notion of **generative capacity**: not strings, but strings with *links* representing dependencies (*derivational* generative capacity)



dat Jan Piet de kinderen zag helpen zwemmen

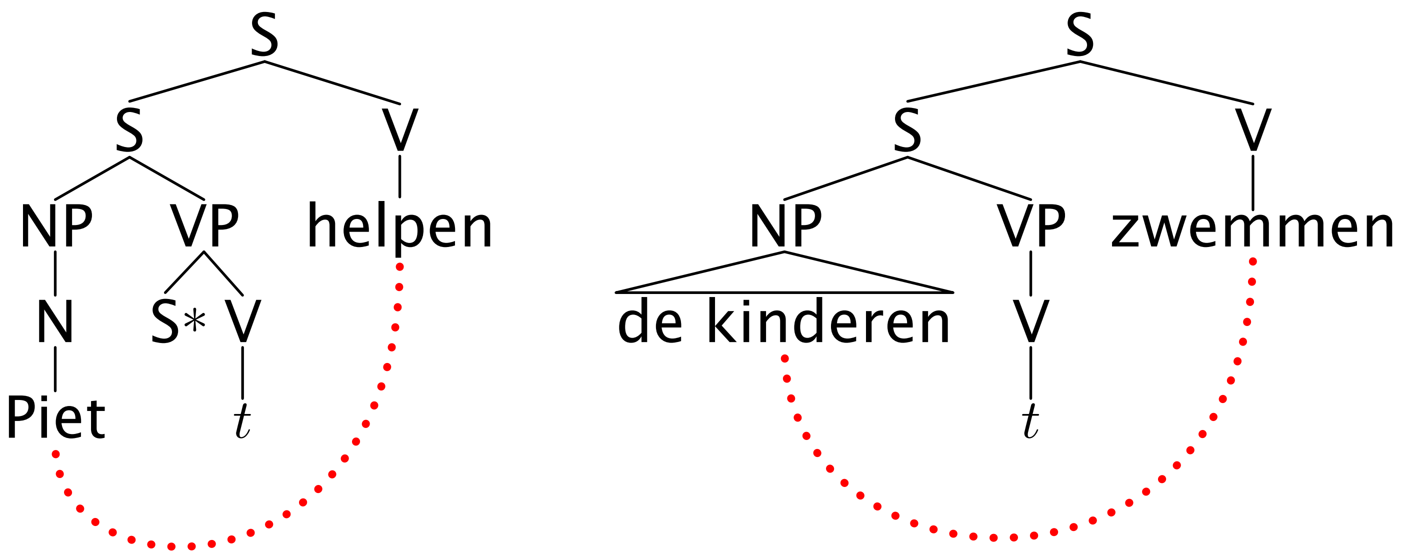
2. A **locality** constraint on how grammars generate these objects: links must be confined to a single *elementary structure*

Historical example: a solution

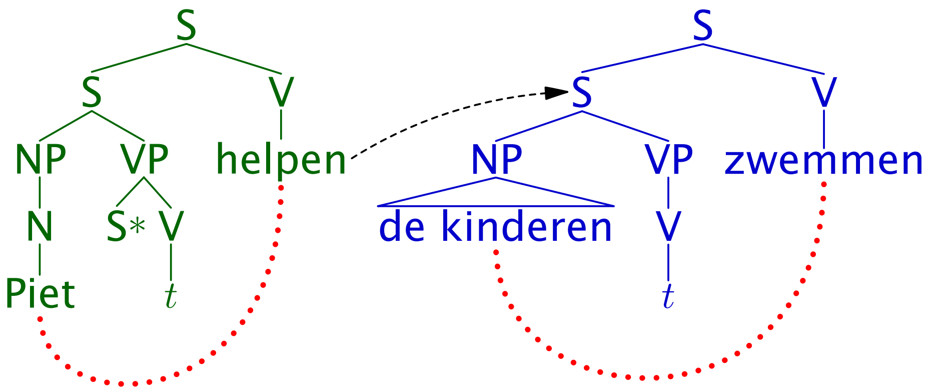
- CFG can't do this

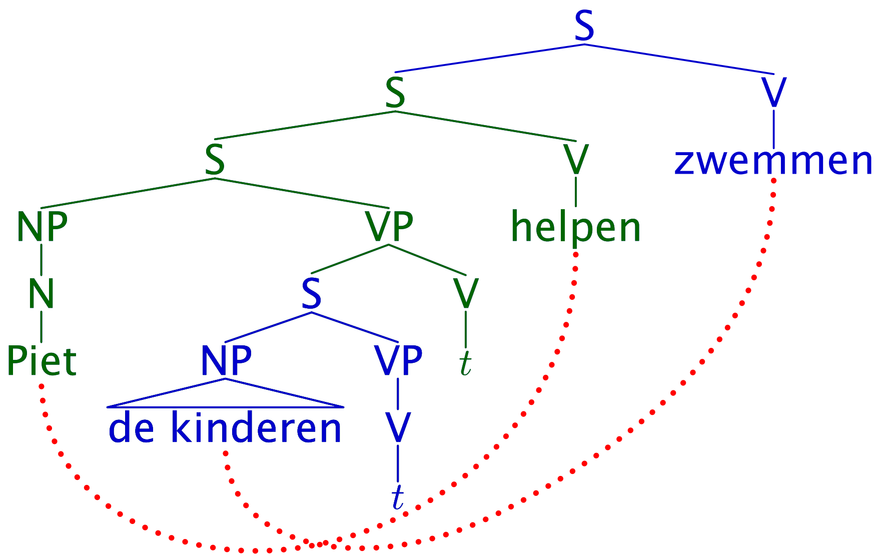
S → Piet S? helpen S?   S → de kinderen S? zwemmen S?

- TAG can



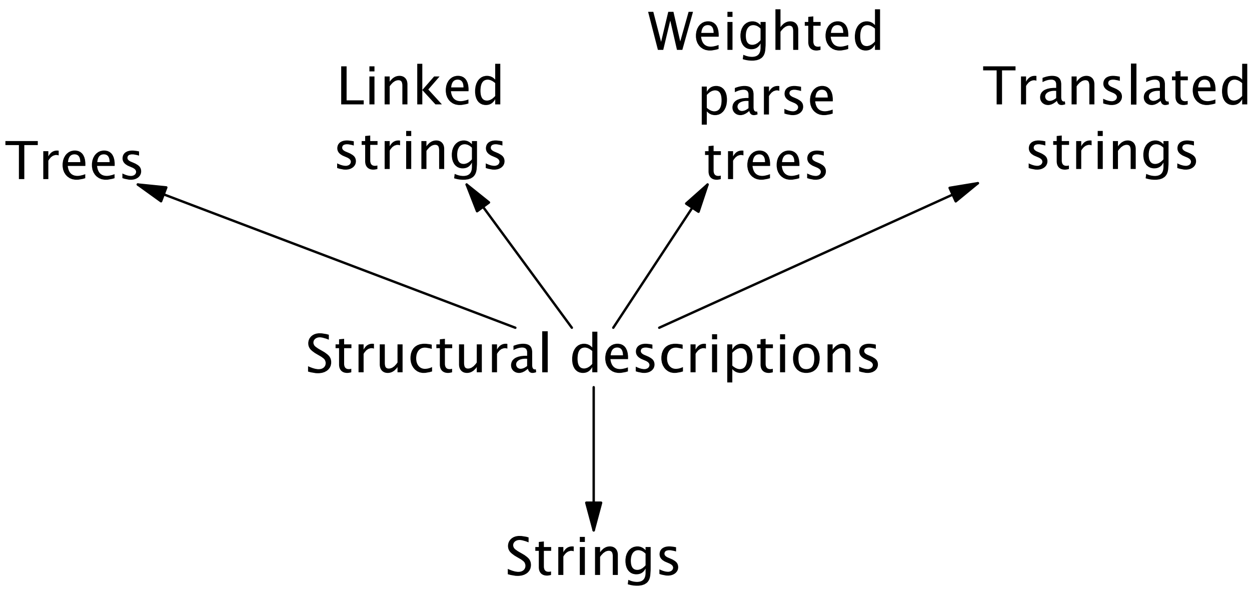
Historical example: a solution





Miller (1999): relativized SGC

- Generalize from DGC to many notions of **SGC**
- Miller: SGC should not compare SDs, but *interpretations* of SDs in various *domains*



Joshi et many al.: Local grammar formalisms

- Generalize from TAG to many formalisms, retaining the idea of **locality**:
  - SDs built out of a finite set of *elementary structures*
  - Interpretation functions factor into *local interpretation functions* defined on elementary structures
- Linear context-free rewriting systems (Weir, 1988) or simple literal movement grammar (Groenink, 1997)

Combined framework

- Choose interpretation domains to measure **SGC** in a sense suitable for applications

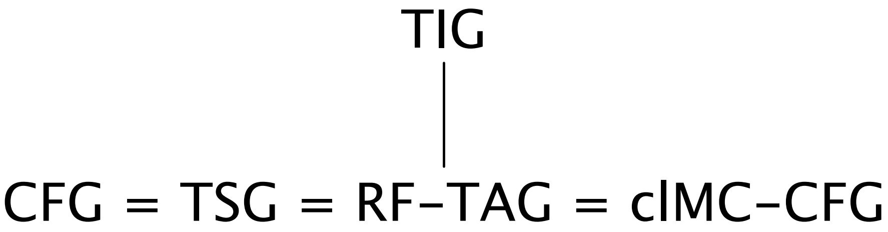
- Define how interpretation functions should respect **locality** of grammars

- Show how various formalisms compare

- Test them by experiments (or thought experiments!)

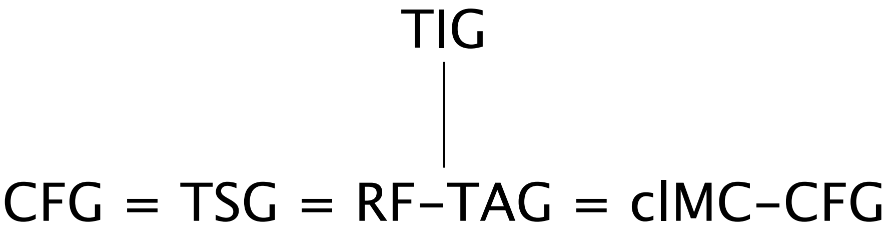
Overview of comparisons: statistical parsing

Trees


TIG

CFG = TSG = RF–TAG = clMC–CFG

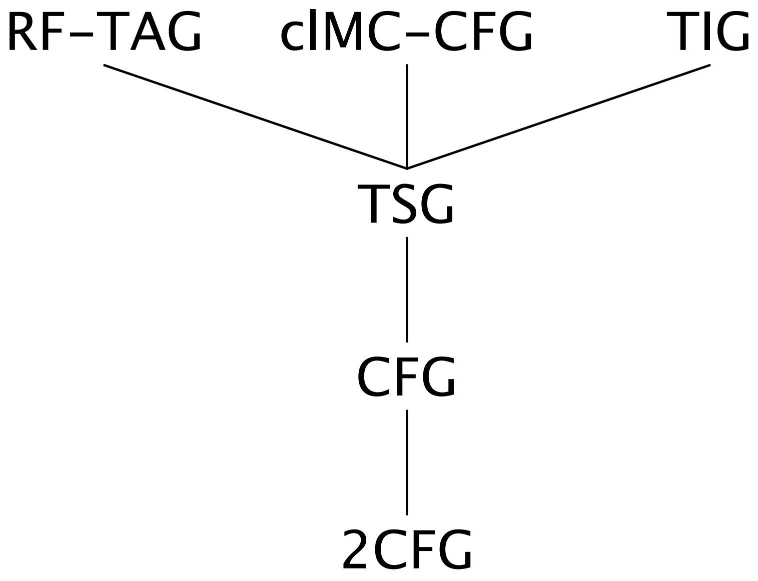
Weighted trees


TIG

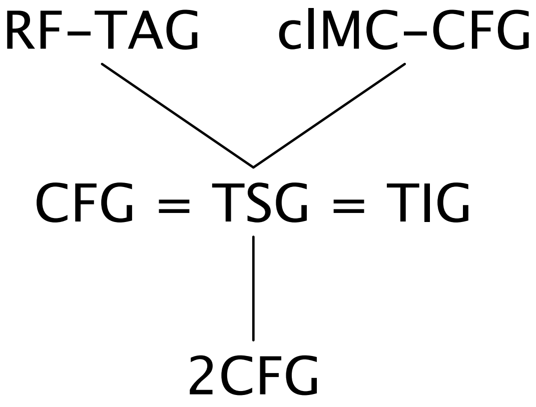
CFG = TSG = RF–TAG = clMC–CFG

Overview of comparisons: translation

Tree relations

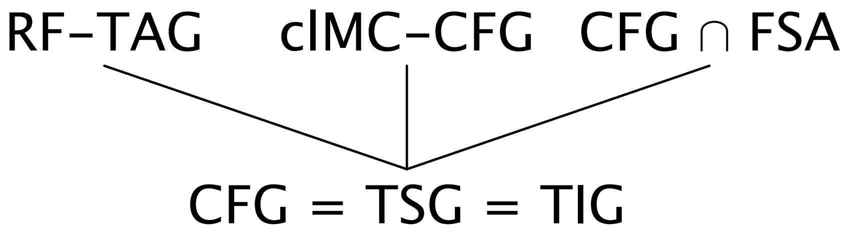


String relations



Overview of comparisons: biological sequence analysis

Weighted linked strings



RF−TAG clMC−CFG CFG ∩ FSA

CFG = TSG = TIG

First application: statistical parsing

- Measuring statistical-modeling power of grammars
- A negative result leads to a reconceptualization of some current parsers
- Experiments on a stochastic TAG-like model

Measuring modeling power

- Statistical parsers use probability distributions over parse structures (trees)

- Statistical parsing *models* map from parse structures to products of parameters
  - History-based: event sequences
  - Maximum-entropy: feature vectors

- Right notion of **SGC**: parse structures with generalized weights

Measuring modeling power

- **Locality** constraint: weights must be decomposed so that each elementary structure gets a fixed weight

- History-based: each elementary structure gets a single event (e.g., PCFG) or event sequence, combine by concatenation

- Maximum-entropy: each elementary structure gets a feature vector (Chiang, 2003; Miyao and Tsujii, 2002), combine by addition

- Grammars with semiring weights

Modeling power for free?

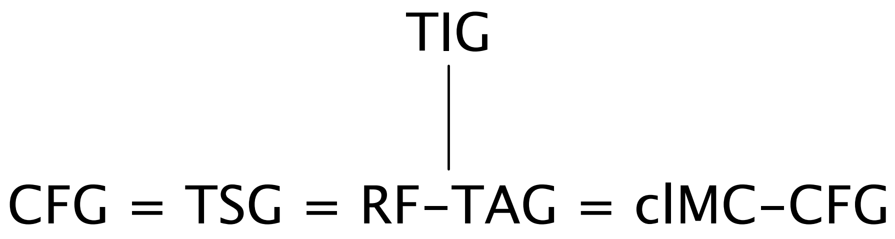
- We might hope that there are formalisms with the same parsing complexity as, say, CFG that have greater modeling power than PCFG

- Often a weakly CF formalism has a parsing algorithm which dynamically compiles the grammar G down to a CFG (a *cover* grammar)

- Easy to show that weights can be chosen for the cover to give the same weights as G

Modeling power for free?

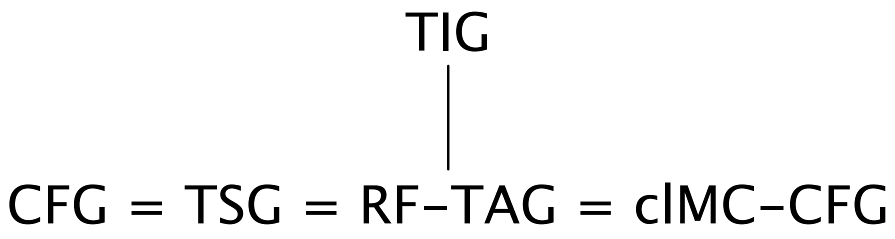
Trees



TIG

CFG = TSG = RF–TAG = clMC–CFG

Weighted trees



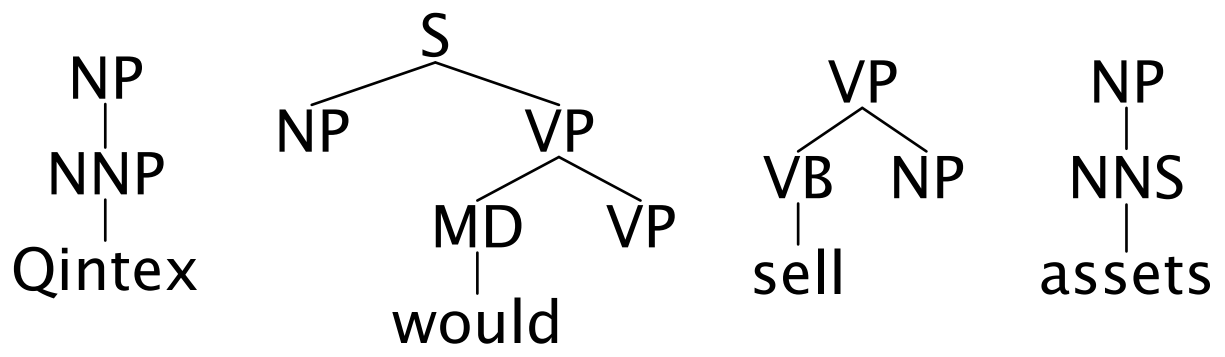
TIG

CFG = TSG = RF–TAG = clMC–CFG

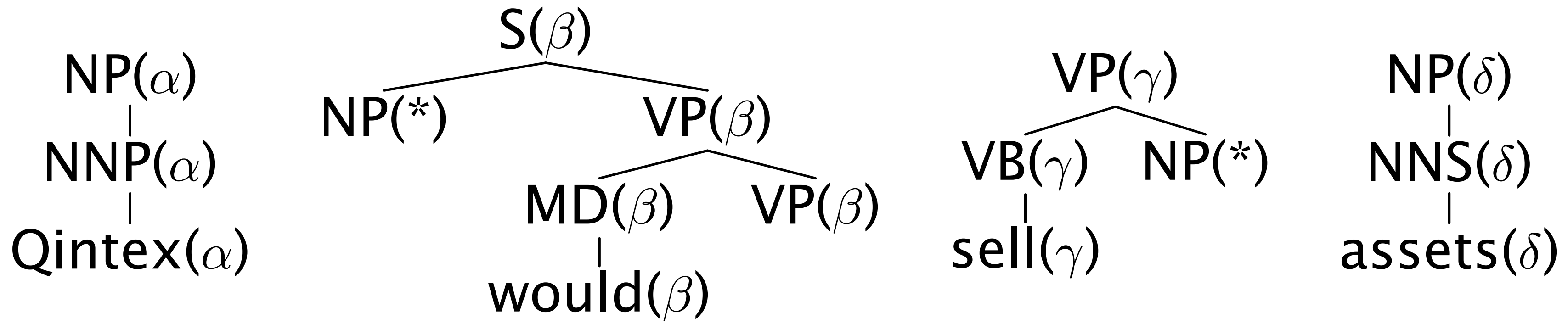
- Not very promising
- However, we may still learn something...

Example: cover grammar of a TSG

- A tree-substitution grammar



- Constructing a cover grammar, step 1:



Example: cover grammar of a TSG

- Constructing a cover grammar, step 2:

$$\begin{aligned}
\text{NP}(\alpha) &\rightarrow \text{PRP}(\alpha) & \text{PRP}(\alpha) &\rightarrow \text{Qintex}(\alpha) \\
\text{S}(\beta) &\rightarrow \text{NP}(*)\ \text{VP}(\beta) & & \\
\text{VP}(\beta) &\rightarrow \text{MD}(\beta)\ \text{VP}(*) & \text{MD}(\beta) &\rightarrow \text{would}(\beta) \\
\text{VP}(\gamma) &\rightarrow \text{VB}(\gamma)\ \text{NP}(*) & \text{VB}(\gamma) &\rightarrow \text{sell}(\gamma) \\
\text{NP}(\delta) &\rightarrow \text{NNS}(\delta) & \text{NNS}(\delta) &\rightarrow \text{assets}(\delta)
\end{aligned}$$

Example: cover grammar of a TSG

- But this is almost identical to the PCFGs many current parsers use

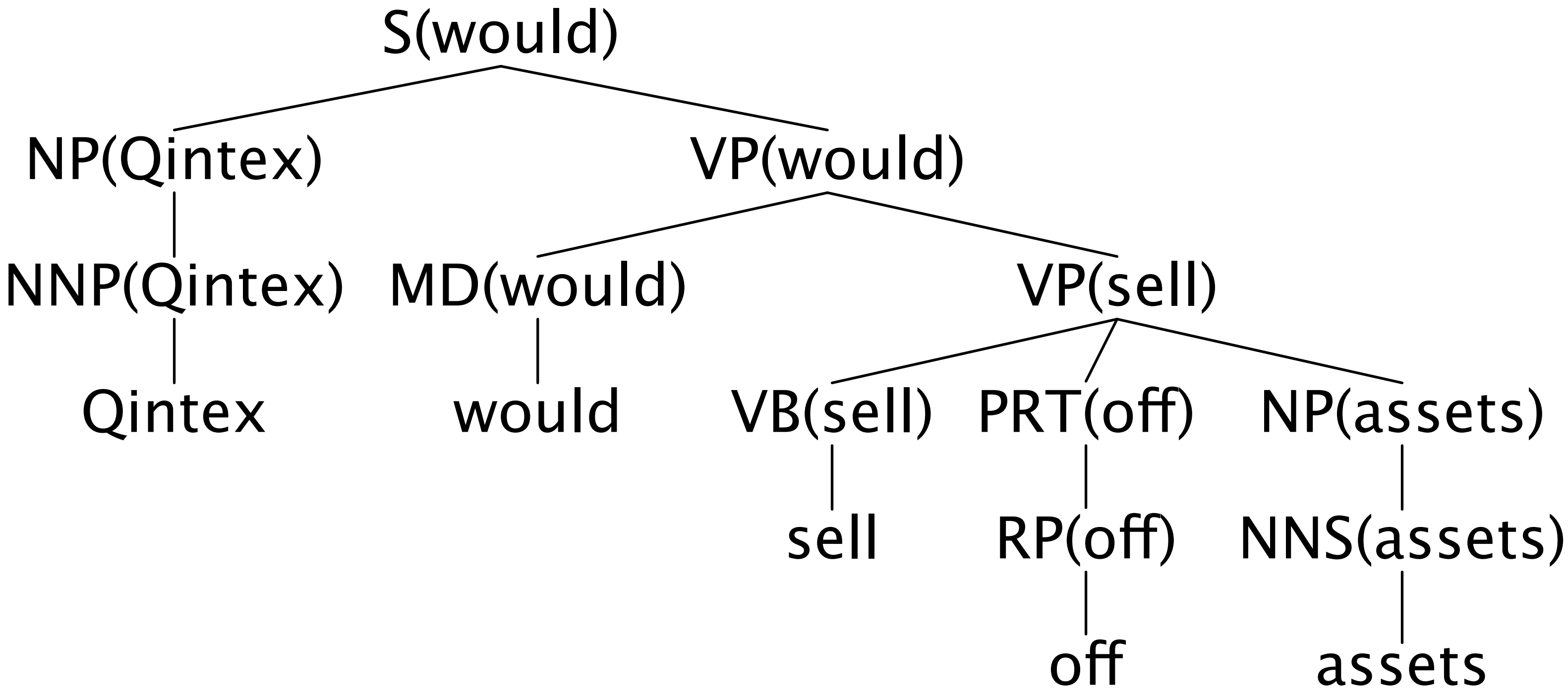
$$\begin{array}{rcl@{\qquad}rcl}
\text{NP(Qintex)} & \rightarrow & \text{PRP(Qintex)} & \text{PRP(Qintex)} & \rightarrow & \text{Qintex} \\
\text{S(would)} & \rightarrow & \text{NP}(*)\ \text{VP(would)} & & & \\
\text{VP(would)} & \rightarrow & \text{MD(would)}\ \text{VP}(*) & \text{MD(would)} & \rightarrow & \text{would} \\
\text{VP(sell)} & \rightarrow & \text{VB(sell)}\ \text{NP}(*) & \text{VB(sell)} & \rightarrow & \text{sell} \\
\text{NP(assets)} & \rightarrow & \text{NNS(assets)} & \text{NNS(assets)} & \rightarrow & \text{assets}
\end{array}$$

(Charniak, 1997, 2000; Collins, 1997, 1999)

- Think of these PCFGs as a compiled version of something with richer SDs, like a TSG

Lexicalized PCFG

Train from the Treebank by using heuristics (head rules, argument rules) to create lexicalized trees



S(would)

NP(Qintex) VP(would)

NNP(Qintex) MD(would) VP(sell)

Qintex would VB(sell) PRT(off) NP(assets)

sell RP(off) NNS(assets)

off assets

Lexicalized PCFG as a cover grammar

- Conventional wisdom: propagation of head words *rearranges* lexical information in trees to bring pairs of words together

- But experiments show that bilexical statistics not as important as lexico-structural statistics (Gildea, 2001; Bikel, 2004)

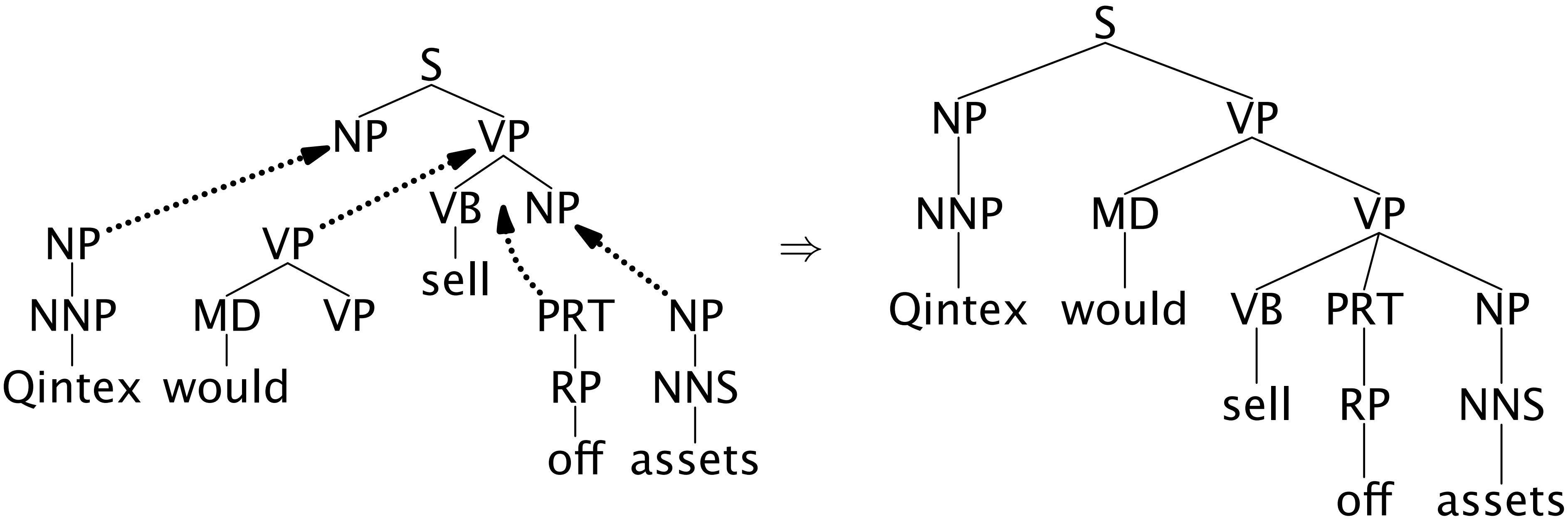
- These structures are in the propagation paths and subcategorization frames

- New view: what matters is the structural information *reconstructed* heuristically

A stochastic TIG model (Chiang, 2000)

- Direct implementation of new view—why?
- Sometimes better not to use head word as a proxy
- Greater flexibility (e.g., multi-headed elementary trees)
- Alternative training method

A stochastic TIG model (Chiang, 2000)



$P_i(\alpha)$ start with initial tree α

$P_s(\alpha \mid \eta)$ substitute α at node η

$P_{sa}(\alpha \mid \eta, i)$ sister-adjoin α under η between ith, $(i{+}1)$st children

$P_a(\beta \mid \eta)$ adjoin β at node η (β's foot node must be at left or right corner)

First training method: extraction heuristics (Chiang, 2000)

- Use heuristics (head rules, argument rules) to reconstruct TAG derivations from training data
- Do relative-frequency estimation on resulting derivations
- Advantages: fast, simple
- Disadvantages:
  - handwritten rules doesn’t always work perfectly
  - relies on reconstructed data

Second training method: EM (Hwa, 1998; Chiang and Bikel, 2002)

- Start with model from previous method
- Iteratively maximize likelihood of *observed* data by Expectation-Maximization
- Advantages: more data-driven
- Disadvantages: slow

Results (English)

Training on WSJ sections 02–21, testing on section 23, sentences ≤40 words

Model	Lab. recall	Lab. precision	F-measure
Rules	87.7	87.8	87.7
Rules+EM	87.2	87.5	87.3
Magerman (1995)	84.6	84.9	84.7
Charniak (2000)	90.1	90.1	90.1

Rules = head rules adapted from Magerman; argument rules from Collins

- Same level of accuracy as lexicalized PCFG
- Reestimation doesn't help

Results (Chinese)

Training on Xinhua sections 001–270, testing on sections 271–300, sentences $\leq$40 words

Model	**Corpus**	**LR**	**LP**	**F**
Rules	Xinhua	78.4	80.0	79.2
Rules+EM	Xinhua	78.8	81.1	79.9
Bikel (2002)	Xinhua	77.0	81.6	79.2
Rules	Xinhua English	76.4	82.3	79.2

Rules = head/argument rules adapted from Xia

- Slightly behind current best parser
- Reestimation seems to edge accuracy past the current best parser

Statistical parsing: conclusion

- Shouldn't hope to get (much) statistical-modeling power for free
- Models like lexicalized PCFG can be thought of as compiled versions of richer models
- Made explicit in a stochastic TIG model with comparable accuracy to lexicalized PCFG models
- Future work:
  - Model and both training methods have room for improvement
  - Maximum-entropy models

Second application: translation

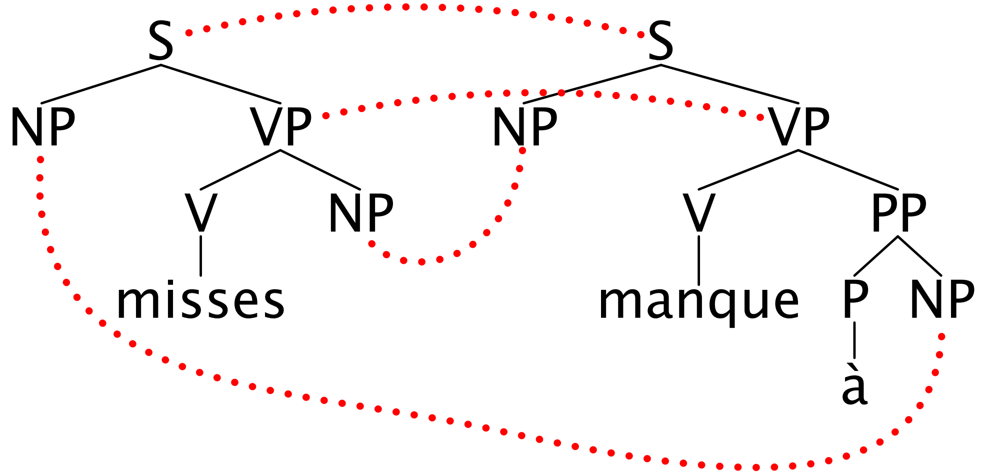
- Measuring translation power of grammars
- Comparing translation power
- Implications for syntax-based machine translation

Measuring translation power

- Right notion of **SGC**: string relations or tree relations
- **Locality** constraint: define mapping on elementary structures
- Synchronous grammar
  - Set of pairs of elementary structures
  - Grammar specifies mapping between paired structures
  - But parallel derivations must be isomorphic

Example: synchronous TAG

- Pairs of elementary structures with linked rewriting sites

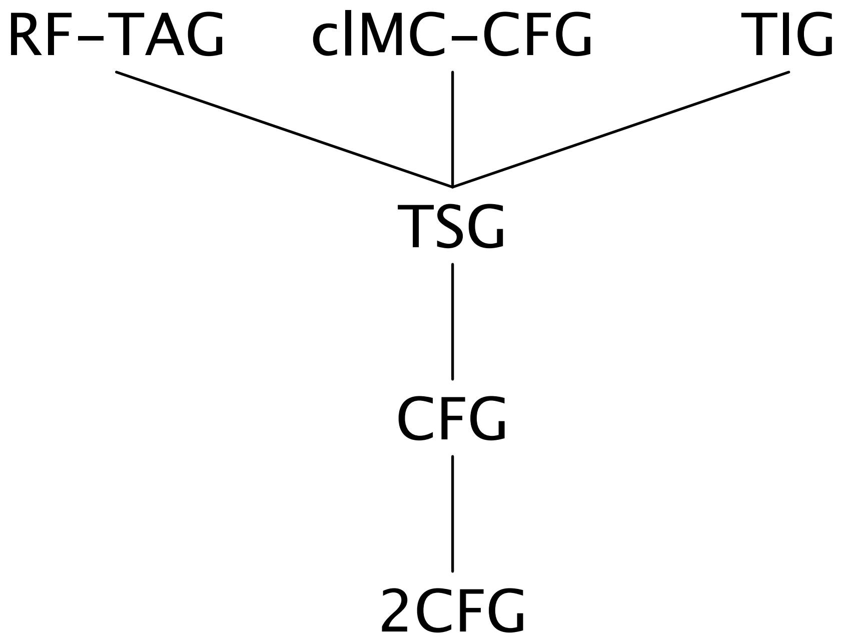


John misses Mary   Mary manque à John

- Rewriting operations take place simultaneously at linked sites

Translation power of various formalisms

Tree relations



RF−TAG

clMC−CFG

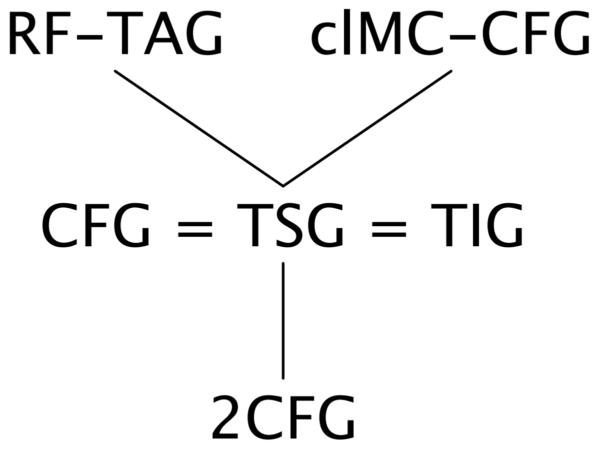
TIG

TSG

CFG

2CFG

String relations



RF−TAG

clMC−CFG

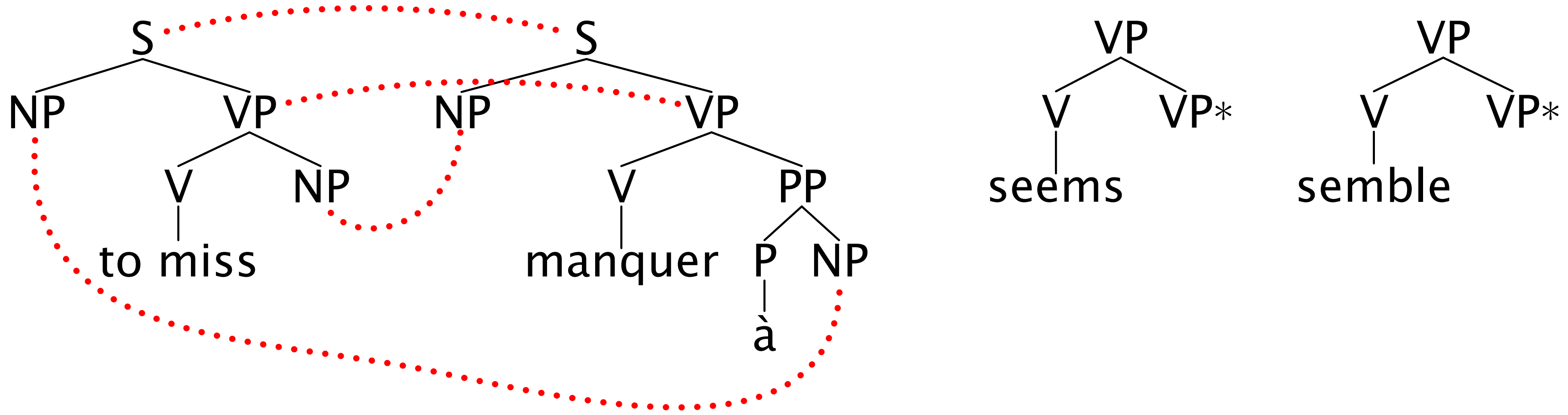
CFG = TSG = TIG

2CFG

Toy example

- RF-TAG: adjunction into middle of spines is restricted (foot unrestricted)

- Synchronous RF-TAG can still “stretch” reorderings



- A double contrast with parsing

Conclusion: statistical parsing vs. MT

- Statistical parsing: we *can* and *should* use CFG to simulate grammars with richer SDs
- Machine translation: we *can't* use CFG to simulate richer grammars, so we *should* use richer grammars
- Synchronous RF-TAG would be a conservative extension of a model like (Yamada and Knight, 2001)
- Greater flexibility without dramatic(?) increase in computation

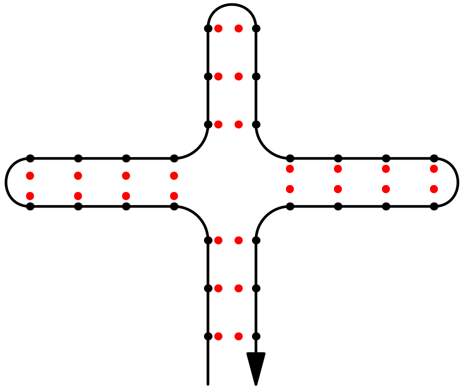
Third application: biological sequence analysis

- Background
- Measuring structure-modeling power of grammars
- Testing extra structure-modeling power

Background: RNAs

- Strings of nucleotides: A, U, C, G

- Bonds form between complementary pairs (A–U, C–G), bending the chain into a *secondary/tertiary structure*:

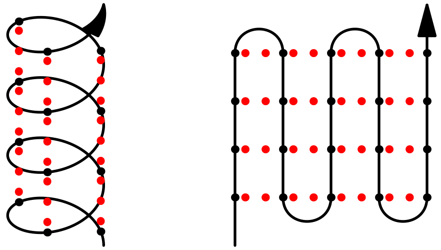


- Messenger RNA is for information storage, but transfer RNA and ribosomal RNA form the machinery used for assembling proteins

Background: proteins

- Sequences of amino acids: 20 types, encoded in triples of DNA bases

- Again, bonds form between amino acids, bending the chain into a secondary/tertiary structure



α–helix β–sheet

- Proteins used for many different purposes: catalyzing reactions, providing physical structure, etc.

Some objectives

- Want to accurately model relationship between sequences and possible structures

- Also want to model dynamics:
  - folding process,
  - transitions under temperature changes,
  - fluctuations from native structure which determine function

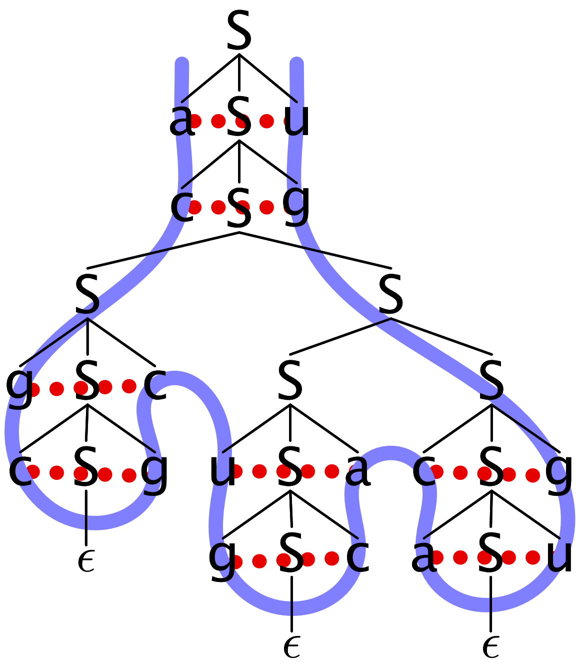
- Potential to improve understanding of biochemical processes

- Potential to facilitate applications like drug design

Grammars for secondary/tertiary structures

- Just as grammars can relate sentences to syntactic structures, maybe they can relate genetic sequences to molecular structures

- Searls (1992): RNA secondary structures ↔ CFG derivation trees

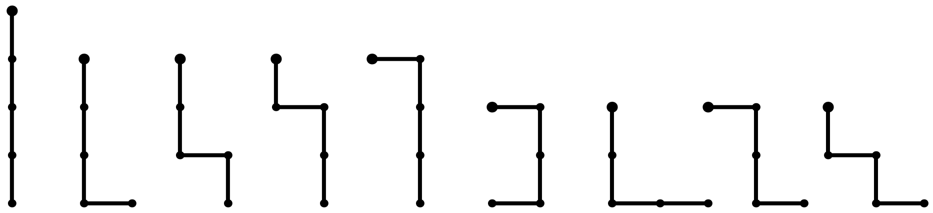


Measuring structure-modeling power

- Right notion of **SGC**: represent folded structures with linked strings
- Moreover, want to model relative importance of structures: weighted linked strings
- Partition function (unnormalized probability distribution)

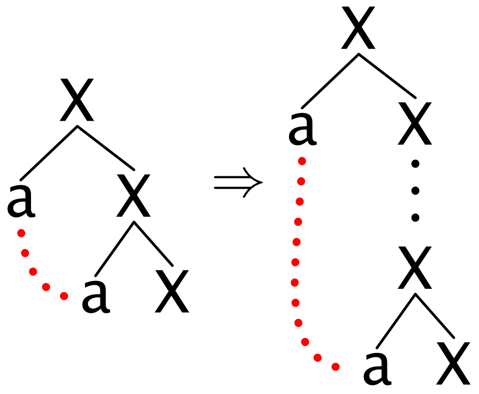
$$Q = \sum_j \Omega_j e^{-E_j/kT}$$

- E_j is energy, Ω_j is number of *conformations*



Grammars for secondary/tertiary structures

- **Locality** constraint: restrict self-contacts to elementary structures
- Generalize beyond CFG; with “stretching” we might lose nice drawings



  but the modeled structure is still the same

- Most previous approaches (informally) follow these principles

Grammars for partition functions

- Decompose term $\Omega_j e^{-E_j/kT}$ into factors $\omega e^{-\Delta E/kT}$, one for each elementary structure

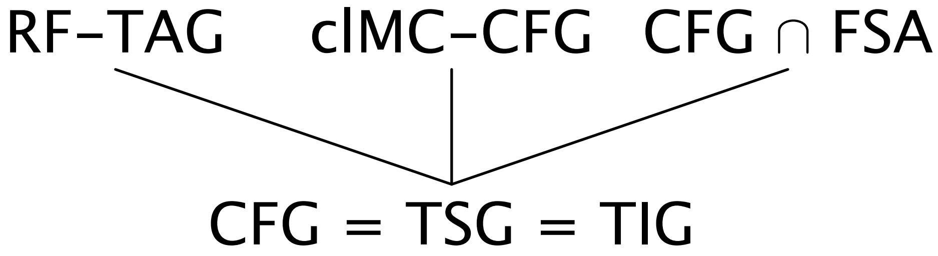
- Grammar must be designed properly
  - energies ΔE should be approximately independent
  - conformation counts ω should be approximately independent

- Then the parser can give us the total Q or various subtotals of Q

- (Chen and Dill, 1995, 1998) as a CFG

Structure-modeling power of various formalisms

Weighted linked strings

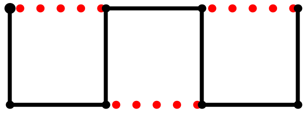


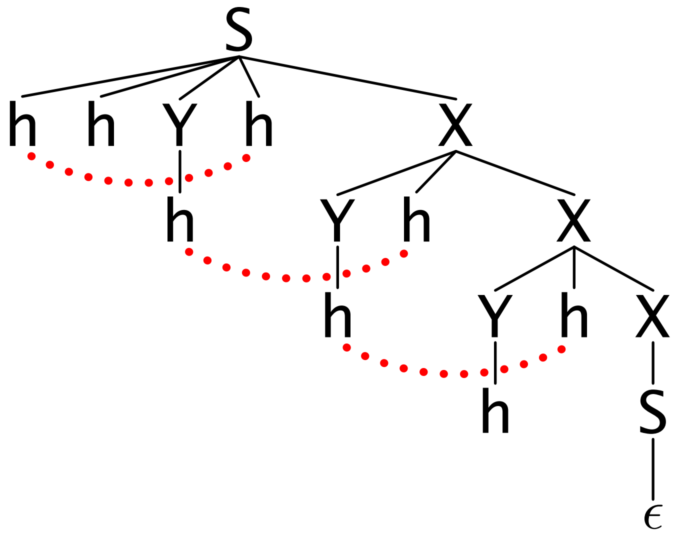
RF-TAG clMC-CFG CFG ∩ FSA

CFG = TSG = TIG

Squeezing DGC out of CFG

- CFG can basically only handle nested dependencies
- RF-TAG and clMC-CFG can handle limited crossing dependencies (Chiang, 2002)
- clMC-CFG: can simultaneously rewrite sister nodes





Intersection

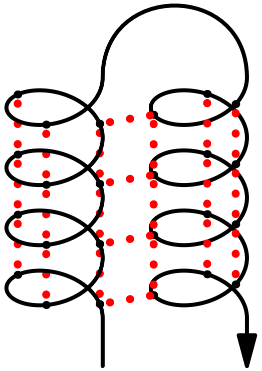
- Idea: analyze a string with two different grammars, or two different parts of a grammar, and merge their SDs

- Largely overlooked in NLP

- For biomolecules: (Brown and Wilson, 1996) tried to intersect CFLs for a type of RNA structure with crossing links, but flawed

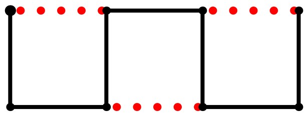
A new problem: helix bundles

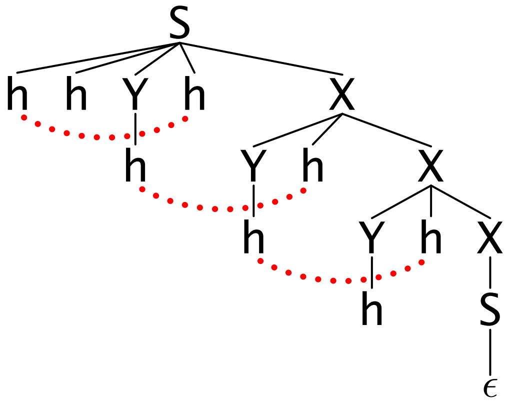
- Chen and Dill’s model captures nested links
- Well–established theory of partition functions of α–helices (Zimm–Bragg)
- Want to combine to form a theory of helix *bundles*



Intersecting a CFG and a finite-state automaton

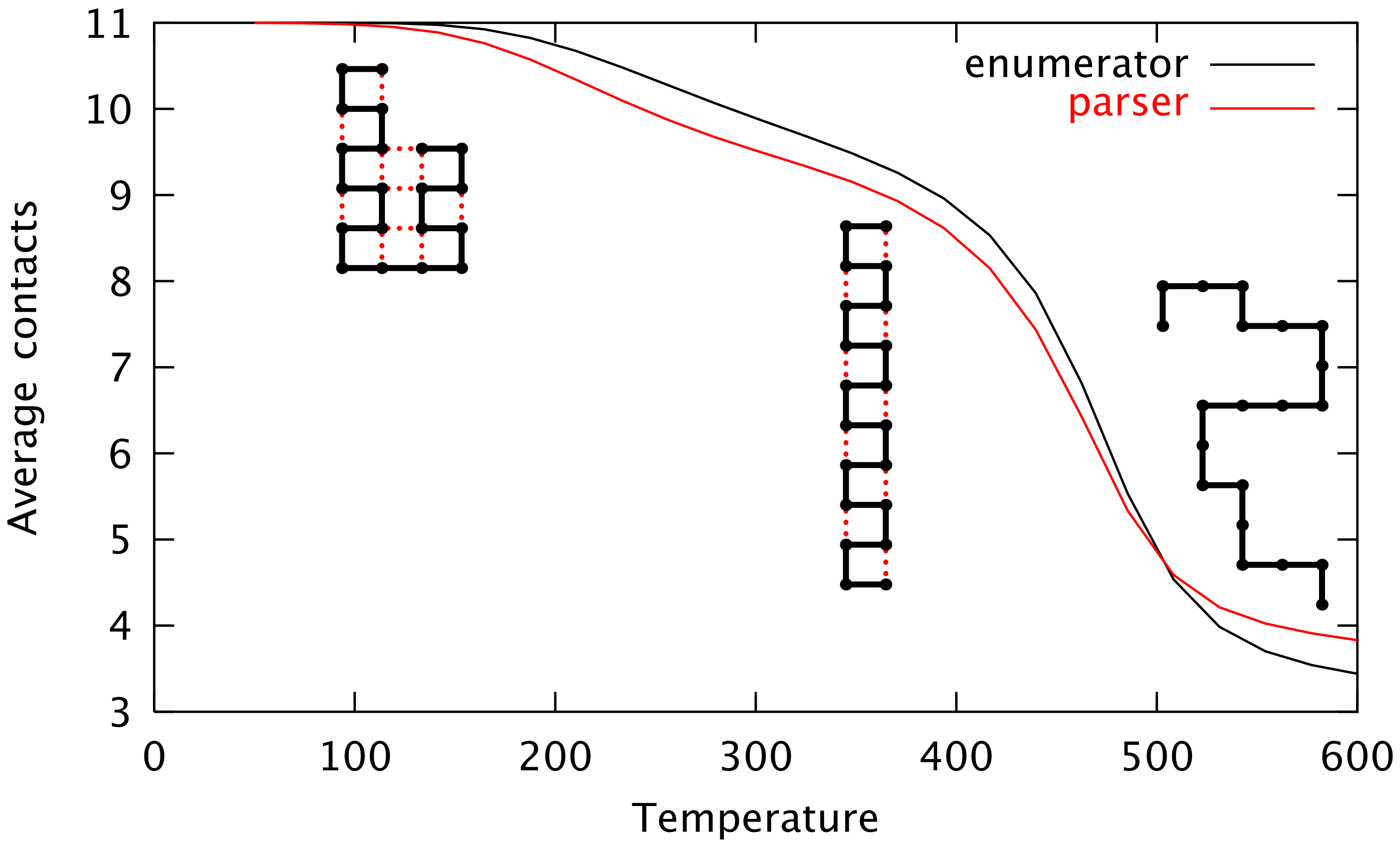
- Chen and Dill's model is a CFG
- α-helices
  - Our grammar is coverable by a finite-state machine





  - Zimm-Bragg (a Markov chain) supplies the weights
- Combine the two by intersection

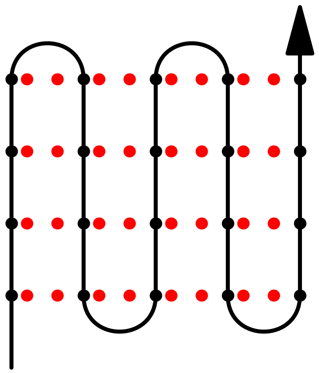
Comparison against exact enumeration



Sequence: hpphhpphhpphhpphhpph

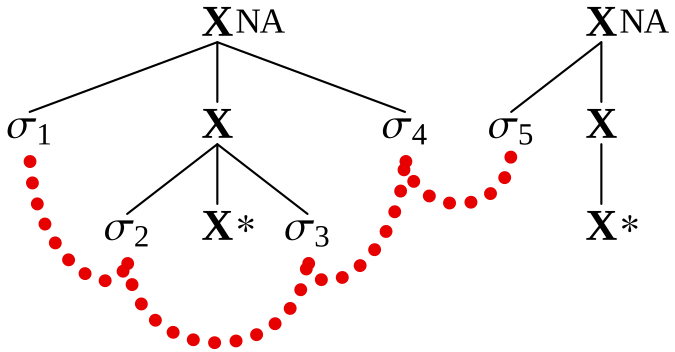
A further problem: larger helix bundles, β–sheets

- Above approach, because based on CFG, can only bundles of two antiparallel helices
- Can we do better?
- Similar to β–sheets



Multicomponent TAG for β–sheets?

- Could use an MC-TAG (Abe and Mamitsuka)



- But parsing complexity is exponential in number of strands
- Prone to spurious ambiguity? (many derivations, one structure)

Simple literal movement grammar

- Closely related to range concatenation grammar (Boullier, 2000)
- Basic idea:

$$\mathsf{S} \rightarrow \mathsf{NP}\ \mathsf{VP}$$

$$\mathsf{S}(xy) \text{:--} \mathsf{NP}(x), \mathsf{VP}(y)$$

- Allows intersection:

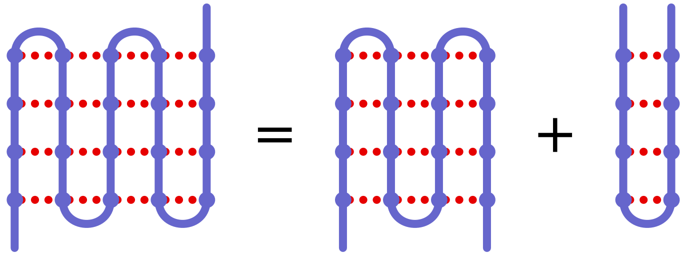
$$\mathsf{A}(x) \text{:--} \mathsf{B}(x), \mathsf{C}(x)$$

- And “partial” intersection:

$$\mathsf{A}(xyz) \text{:--} \mathsf{B}(x, y), \mathsf{C}(y, z)$$

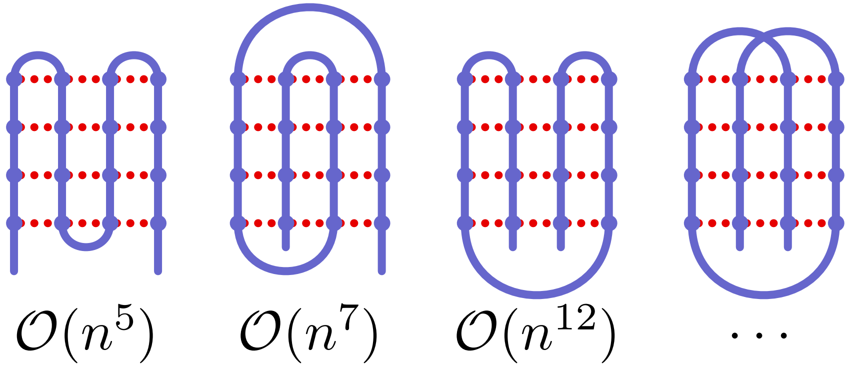
An sLMG analysis of β−sheets

- Generating pairs of antiparallel strands (hairpin) or parallel strands is easy
- Use intersection to combine them into a sheet
- Essentially, build a sheet by merging last strand of a sheet with one strand of a hairpin



An sLMG analysis of β–sheets

- Faster than MC–TAG analysis ($\mathcal{O}(n^5)$ for any number of strands)
- Permuting the strands makes complexity go up, no advantage in worst case



- Computational complexity seems to correlate with folding difficulty
- Certain inter–hairpin dependencies could make the problem NP–hard

Biological sequence analysis: conclusion

- Synthesized and formalized existing approaches
- Recast Chen and Dill's model as a weighted CFG, opening the door to richer models
- Limited crossing dependencies can be modeled by clMC-CFG or RF-TAG without any extra cost
- Intersection allows modeling of helix bundles and maybe β-sheets

Conclusion

- What makes one grammar formalism better than another? Introduced machinery for giving rigorous answers
- Demonstrated a new view of recent statistical parsers as compiled versions of grammars with richer SDs
- Argued that machine translation stands to gain much more from richer grammars
- Synthesized previous grammatical models of biomolecules and demonstrated some new approaches

Future work

- Statistical parsing: maximum-entropy models
- Translation: implement an RF-TAG version of some existing CFG model
- Biological sequence analysis: extend CFG parser, compare MC-TAG analysis to sLMG analysis
- New application areas