
Thank you very much John. On the one hand I was getting more and more embarrassed as you went on. On the other hand I felt if you kept going like this I will not have to say much afterwards as I will be out of time. But John made sure this will not happen. It is indeed a great honor to receive this award from acl that is from my colleagues and friends in computational linguists. I am very grateful to you for your honoring me in this way. Frankly I feel overwhelmed. John told me I need to make a speech. Well, what does one say at such an occasion. John said I should say something about the current state and future of CL etc.—something like that. That ruled out the possibility of using all the time to talk about whatever I am doing at present—that would not be quite right. One could try to collect all the papers rejected in the past, especially by ACL and try to summarize them on this occasion. That would not be cricket! Talking about the current state and future etc. Well, talking about such things at a very high level does not really take much time. General predictions are easy to make and specific predictions turn out to be wrong, in any case. One could try to give some advice, advice is cheap. But as we all know advice is best ignored as one's graduate students do it all the time. Anyway, having given some thought to all these considerations and John's comments I have prepared some comments partly based on some of my own work and partly concerning some general issues.

So here are –Some Random Thoughts over a Lifetime—slide..

SOME RANDOM THOUGHTS OVER A LIFETIME

ARAVIND K. JOSHI

Department of Computer and Information Science

and

Institute for Research in Cognitive Science

University of Pennsylvania

Philadelphia PA, USA

July 8 2002

lifetime

- **Webster:**
 - the time a life continues: the duration of a living being or a thing
- **OED**
 - the time that life continues
- **American Heritage:**
 - the period of time during which an individual is alive
 - the period of time during which an object, property, process, or phenomenon exists or functions

lifetime in Penn Treebank

- There are only 10 sentences containing *lifetime*
- Here is a typical one

I always knew that the Big One was coming,
but not during my **lifetime**, she says.

- Clearly here *lifetime* means *lifetime*
- However, there is another group of sentences such as

lifetime in Penn Treebank

Mr. Thompson played outfield and third base until 1960, posting a lifetime .270 batting average and chalking up 264 home runs before retiring and going into paper-goods sales.

- *lifetime* can mean less than *lifetime*

Outline

- **Finite state transducers for parsing**
- **Relationship among formal/computational systems**
- **Relationship between Linguistics and CL**
- **Selling paper-goods!**

Finite State transducers (FST) for parsing

- A program developed at the University of Pennsylvania, 1958-59
- **First FST application to parsing**
- **Recently reconstructed from original documentation, renamed Uniparse, 1996 and evaluated on very small subsets of corpora – WSJ, IBM computer manuals, ATIS!**

Finite State transducers (FST) for parsing

- **Original participants**
 - **Lila Gleitman, Aravind Joshi, Bruria Kauffman, Naomi Sager, and Carol Chomsky**
 - **Overall project (Transformations and Discourse Analysis Project) directed by Zellig Harris**
- **Reconstruction from original documentation**

Joshi and Hopely. 1998. A Parser from Antiquity, in *Extended Finite State Models of Language* (ed. A. Kornai), Cambridge University Press

 - **comments by Lauri Karttunen in the same volume**

Finite state computations

- **Cascaded finite state transducers (fst) for computing**
 - dictionary look-up and grammatical idioms
 - part-of-speech disambiguation
 - simple noun phrases
 - simple adjuncts – prepositional phrases, adverbial phrases
 - verb clusters
 - clauses (strictly speaking not an fst computation)
- Partial parsing – attachments are not shown explicitly

Uniparse – an example

[We] { have found } / that [subsequent addition]
(of [the second inducer]) (of [either system])
< after { allowing } [single induction] { to proceed }
+ > (for [fifteen minutes]) (also) { results } (in
[increased production]) + \ + (of [both enzymes])

[] simple noun phrases, () simple adjuncts, { } verb clusters
< > clauses, / \ clauses. + end of a complement

Finite State Computation

- **Current situation**
 - **finite state calculi**
 - **enormous sizes of finite state transducers**
 - **fast determinization and minimization techniques**
 - **stochastic finite transducers**

Uniparse—retrospective comments

- **Why did the work on cascaded fst not continue?**
 - **Growing sizes of fst's, extremely limited computing resources**
 - **No systematic way of backtracking**
 - **No systematic ways of minimizing and determinizing fst's**
- **A new technique comes out of an application but then does not go further. This happens very often and marks the beginning of theoretical work, e.g. left to right parsing, CFG parsing, string and tree grammars, etc., -- and then the old technique is rediscovered!**

Uniparse—retrospective comment

- **This is a case study of how and why certain lines of work stop and then get rediscovered often several years later -- perhaps ask students in intro CL courses to look at old literature and reconstruct some old systems!**
- **This might shorten the period of rediscovery**
 - **It may also help give a better historical sense of the field**

Relationship among formal/computational systems

- **Constrained formal/grammatical systems**
 - **Tree-Adjoining Grammars (TAG)**
 - **Linguistic aspect: extended domain of locality**
 - **Computational aspect: factoring recursion from the domain of dependencies**
 - **Processing aspect: automaton equivalent of TAG-EPDA**
 - **70's – early 90's**
 - **Linguistic, computational, and processing properties of TAG and its variants, MCTAG LCFRS, Description Tree Grammars, etc.**

Relationship among formal/computational systems

- **Tree-Adjoining Grammars**
 - **70's – 90's – continued**
 - **Equivalence of TAG, HG, LIG, CCG**
 - **Compilation of other grammar formalisms into TAG and in the reverse direction also, e.g., HPSG, LFG, versions of GB and minimalist grammars (Kasper et al., Kameyama, Frank, Stabler)**
 - **90's – This type work still continues**

Relationship among formal/computational systems

- **How much of this kind of work can be or should be done?**
- **This sort of work or for that matter most formal work is bootlegged or piggybacked!!**
- **This situation will continue, at least in the near future, as CL is largely driven (and perhaps justifiably so) by immediate and potential applications**

Relationship among formal/computational systems

- In engineering very often new formal techniques are developed and then they become objects of formal study, e.g., the theory of Laplace transforms
- Relationship of CL and formal/mathematical work can be and should be of this kind
 - **Is it the case?**

Relationship among formal/computational systems

- **What is the value of this kind of work?**
 - **For some there may be very little**
 - **If one is interested in CL contributing to the understanding of the structure of language then it has great value in my judgment**
- **Showing equivalence among different systems is considered of great value in many scientific enterprises, as it reveals the invariances**
 - **Analogy to study of different coordinate systems**
 - Cartesian, Polar, ...**

Relationships among formal/computational systems

- **Analogy to coordinate systems– continued**
 - **conversion from one system to another**
 - **some problems are easier to formulate and solve in one system than in another**
 - **use of different coordinate systems for different problems is very common in math/physics/engineering**
 - **Not in CL, why?**

Relationships among formal/computational systems

- **Analogy to coordinate systems – continued**
 - **Computational linguists are very fond of their own systems, much like the linguists!**
 - **CL involves building large resources and therefore too much effort and time may be involved in conversion**
 - **But this need not be the case if there are reliable and efficient conversion packages**

Relationships between Linguistics and CL

- **Various perspectives**
 - **Linguistics (Theory), CL (Applied)– Theory/Applied**
 - **Does linguistics inform CL?**
 - Early 60's to 80's -- yes**
 - 90's to present – yes, maybe, does not matter,
good for annotations, etc.**
 - 2022 -- ?**

Relationships between Linguistics and CL

- **Various perspectives – continued**
 - **Does CL inform linguistics?**
 - 60's – perhaps?**
 - 70's, 80's early 90's – yes, at least some computational linguists thought so**
 - others did not care**
 - Linguists were hesitant to accept the importance of CL**

Relationships between Linguistics and CL

- **Various perspectives – continued**
 - **Does CL inform linguistics?**
 - 90's to present -- yes, at least some computational linguists think so
 - **Now linguists are more open to CL**
but a lot of CL is moving away from linguistics
because of the success of statistical/ml techniques
applied to corpora, annotated (with very little
linguistic information) or, especially, unlabeled
data

Relationships between Linguistics and CL

- **Immediate future: at least two directions**
- **More richly annotated corpora**
 - **more expensive, smaller sizes**
 - **techniques for combining with unlabeled data**
 - **Not sure how far these techniques will scale up, especially for complex annotations**
- **New ways of working with unlabeled data with minimal linguistic information**

Relationships between Linguistics and CL

- **When it comes to discourse**
 - **there is more chance of closer ties between CL and linguistics**
 - **On both sides there is much less work as compared to syntax and semantics**
 - **ignorance on both sides may help them to come together more easily**
- **In general, CL would have more impact on linguistics if CL helps in **discovering new facts about language** because that is what linguistics is supposed to be about and not just about different ways of organizing known facts**

NLP techniques for modeling biological sequences

- **There is already considerable work in this area and some in the reverse direction also**

**Biological Sequence Analysis by Durbin et al.
Cambridge University Press, 1998/2000**

**Time Warps, String Edits, and Macromolecules by
Sankoff and Kruskal, CSLI 1999 with an Introduction
to the reissue edition by John Nerbonne**

Relevance of structural descriptions to modeling

- Sequences are made from an alphabet of 4 nucleic acids (A, C, G, T/U) for DNA and RNA sequences or 20 amino acids for protein sequences
 - Primary structure of sequences – **Linear structure**
 - Secondary structures
 - Tertiary structures
 - Quaternary structures
- ← **Folded structures**
- Folding arises because certain dependent elements have to be spatially adjacent

The black **cat** gracefully **sat** on the old **mat**

RNA secondary structure: Pseudoknots

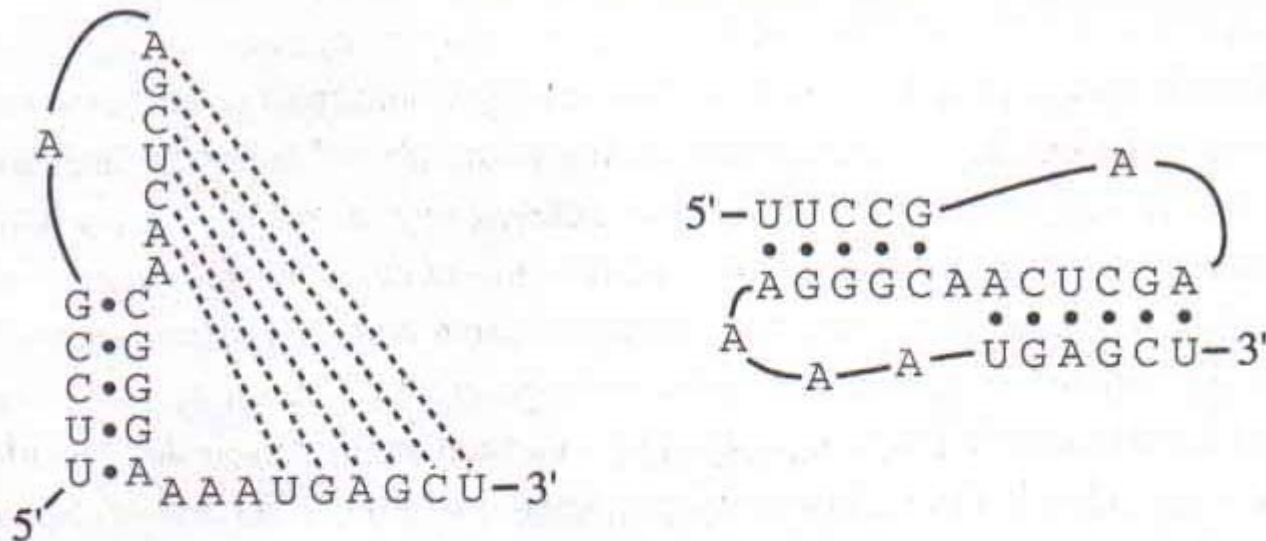
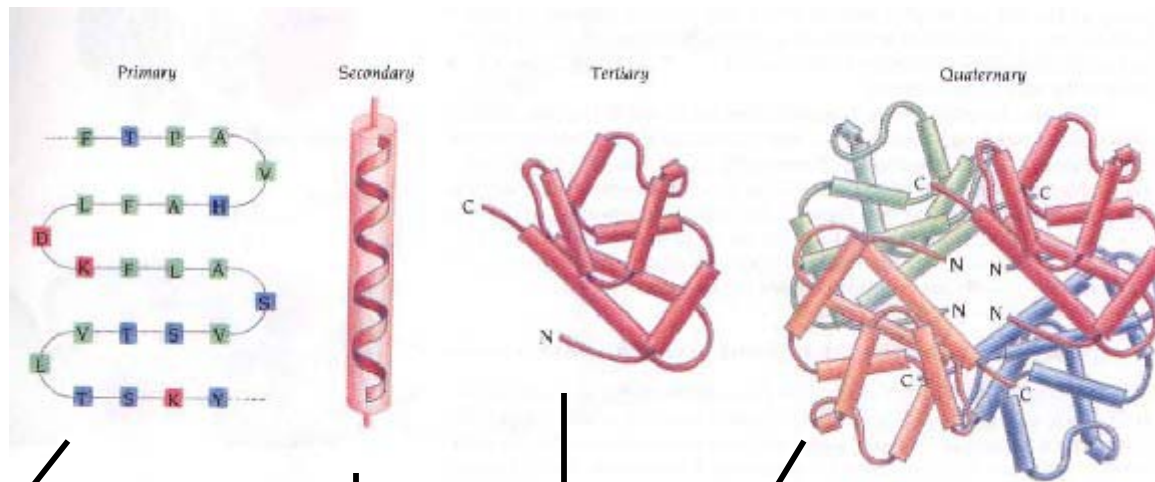


Figure 10.3 Base pairs between a loop and positions outside the enclosing stem are called a pseudoknot (left). Another representation of the same pseudoknot is shown on the right. In three-dimensional space, the two stems can stack coaxially and mimic a contiguous A-form helix. This particular example is an artificially selected RNA inhibitor of the human immunodeficiency virus reverse transcriptase [Tuerk, MacDougal & Gold 1992].

Polypeptide chains: Proteins

Primary Secondary Tertiary Quaternary



Linear sequence
of amino acids

α helices

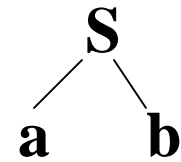
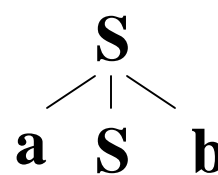
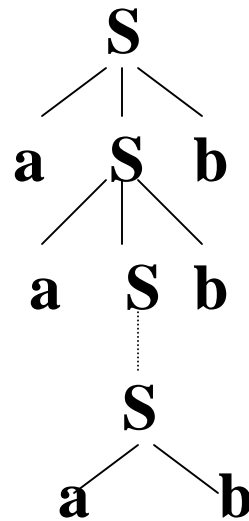
Domains: β sheets connected
by loops and α helices

Multiple connected domains

Nested dependencies

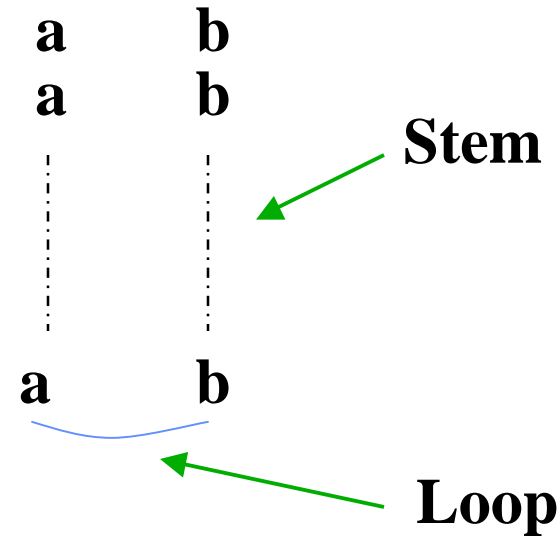
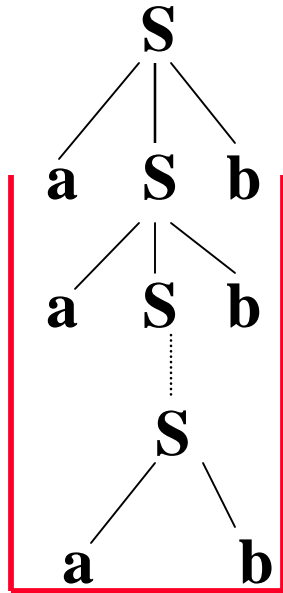
Nested dependencies described by the derivation structure of a grammar, e.g., a context-free grammar (CFG)

G: $S \rightarrow a S b$
 $S \rightarrow a b$



a, b: matching pair
for RNA: (A,U), (C,G)
Proteins: pair of amino acid residues

Structural descriptions and folded structure



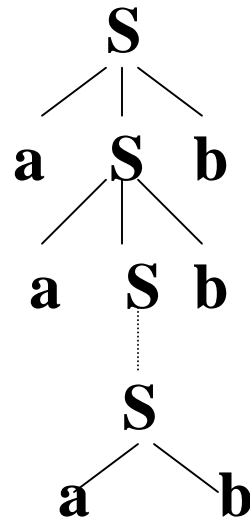
- **Structural description related to the folded structure**
 - direct relationship in this particular case
- **Hairpin structures and some related structures**

Searls 1995, 1999

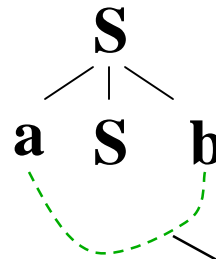
Nested dependencies and adjacencies

Nested dependencies and adjacencies

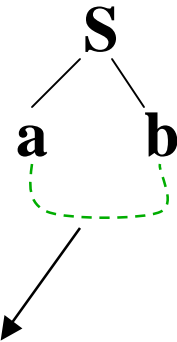
- specified on the elementary structures of a grammar
- elementary trees of TAG



G: β :



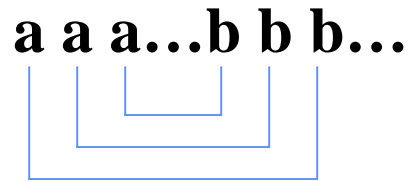
α :



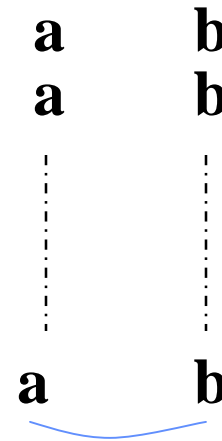
Adjacency constraints

Nested dependencies

-- in the linear and the secondary structure



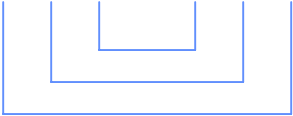
Linear structure



Secondary structure

- **Nested Dependencies : Non-CFG representation**

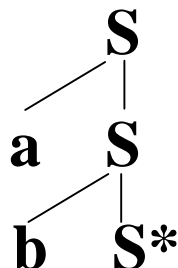
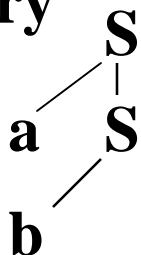
a a a...b b b... $a^n b^n \quad n > 0$



The CFG method of assembling the elementary structures is not the only way to get nested dependencies

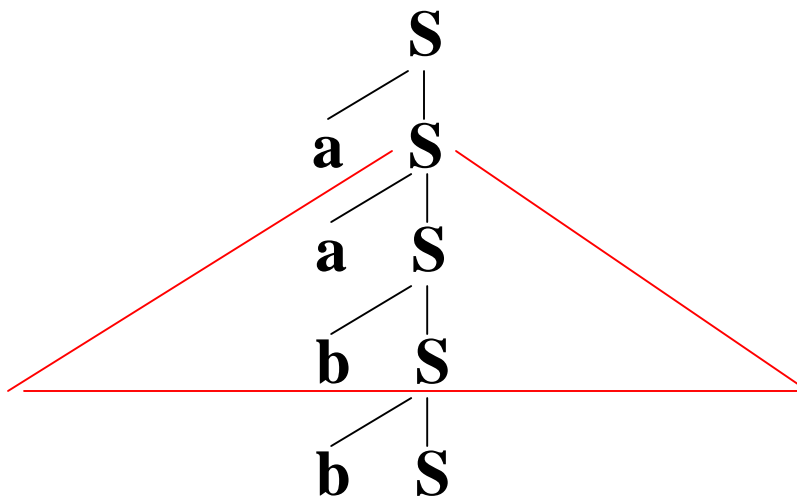
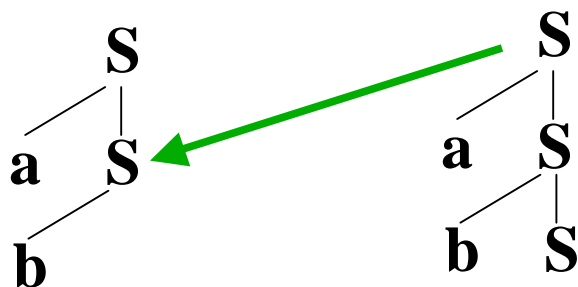
Nested dependencies on the same side of the spine

Elementary trees:

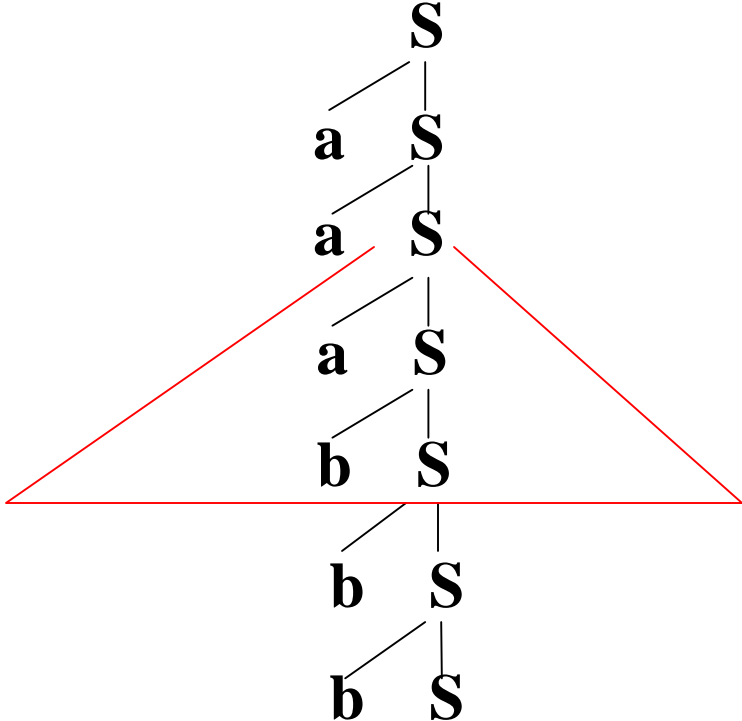
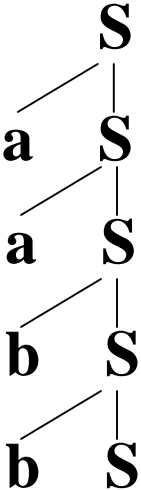


Assembly by
-- substitution
-- adjoining (splicing in)

Derivation:

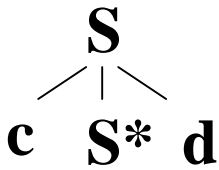
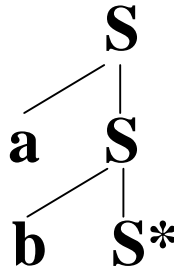
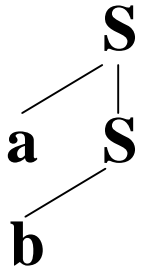


Nested dependencies on the same side of the spine



Pseudoknot

Elementary trees:



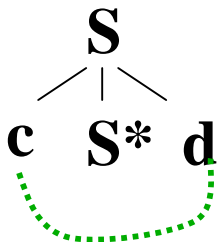
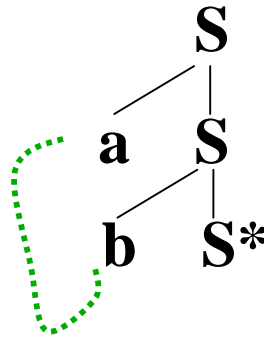
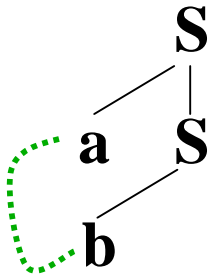
Assembly by

-- substitution

-- adjoining (splicing in)

Nested dependencies on the same side of the spine: Generated by TAG

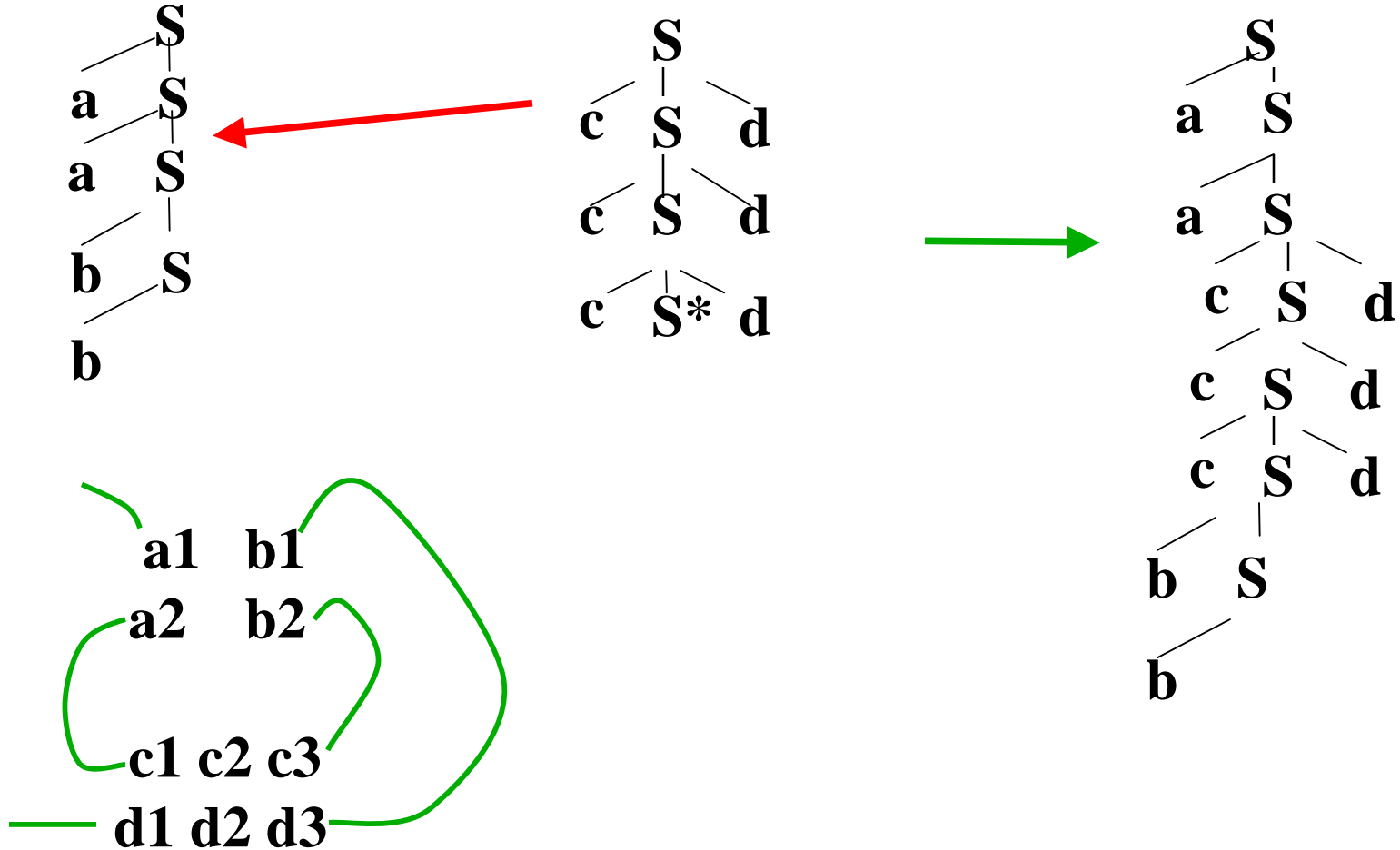
Elementary trees:



Assembly by
-- substitution
-- adjoining (splicing in)

The dotted lines represent the
spatial adjacencies

Pseudoknot



RNA secondary structure: Pseudoknots

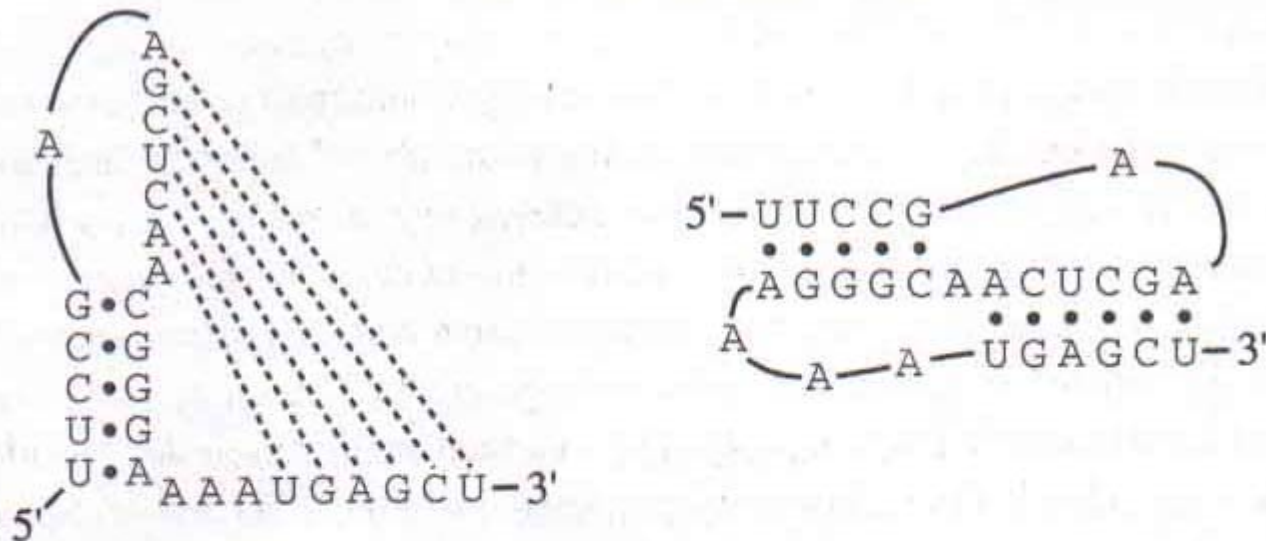


Figure 10.3 Base pairs between a loop and positions outside the enclosing stem are called a pseudoknot (left). Another representation of the same pseudoknot is shown on the right. In three-dimensional space, the two stems can stack coaxially and mimic a contiguous A-form helix. This particular example is an artificially selected RNA inhibitor of the human immunodeficiency virus reverse transcriptase [Tuerk, MacDougal & Gold 1992].

Pseudoknots

Y. Uemura , A. Hasegawa, S. Kobayashi, and T. Yokomori. 1999. Tree-adjointing grammars for RNA structure prediction. *Theoretical Computer Science*, 10:277-303.

(Used a special case of TAGs for modeling pseudoknots)

Pseudoknots

E. Rivas and S. Eddy. 2000. The language of RNA: a formal grammar that includes pseudoknots. *Bioinformatics*, 16(4):334-340.

(Used crossed interaction diagrams-- Feynman Diagrams, with some constraints

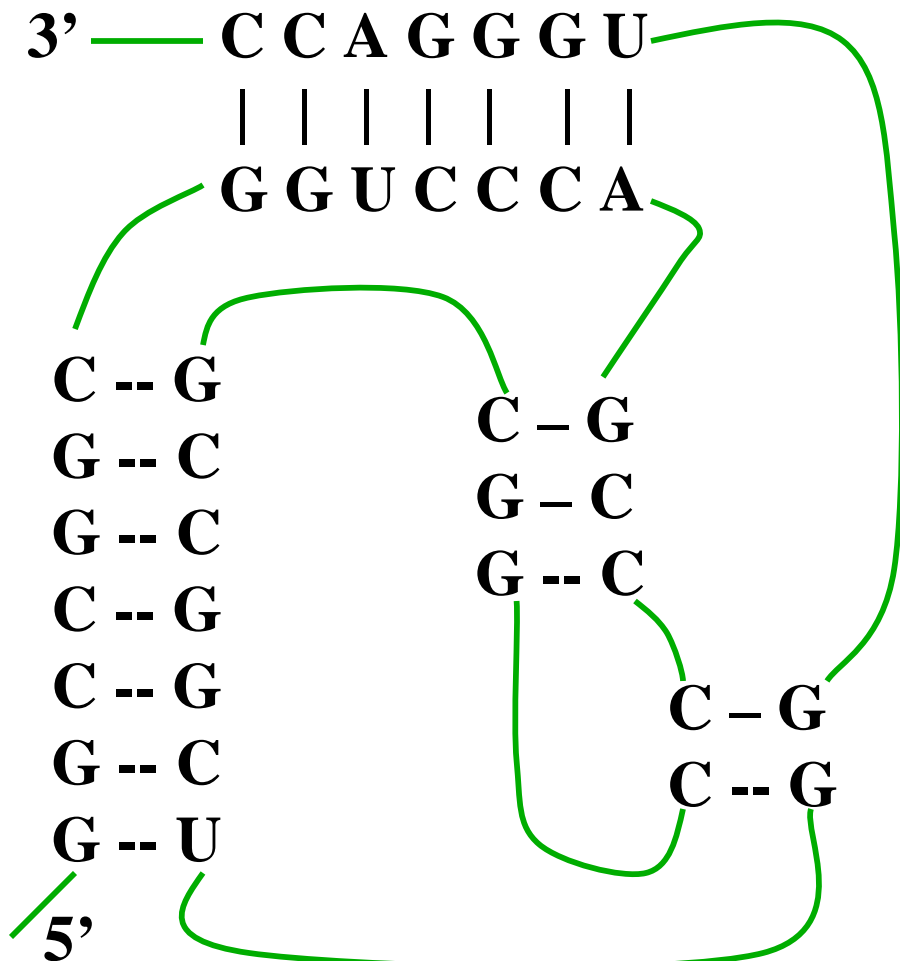
With these constraints, the machinery used by Rivas and Eddy is **no more powerful than TAG or some of its generalizations such as the multi-component TAG, Chiang and Joshi (2001/2002)**

Doubly nested pseudoknot

The most complicated pseudoknot elucidated thus far
-- delta virus (HDV) ribozyme

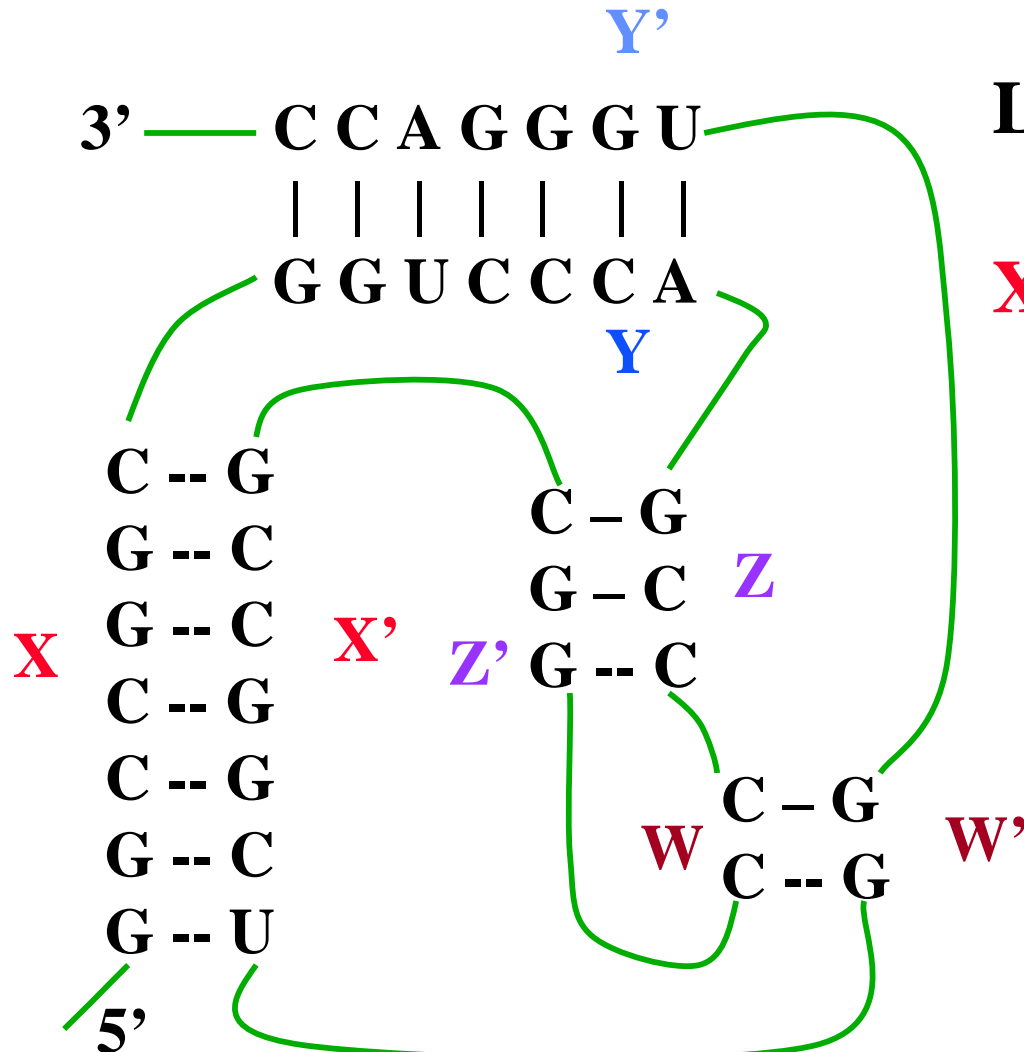
C. W. Hilbers, P. J. A. Michiels, and H. A. Heus. 2000
New developments in structure determination of
pseudoknots. Biopolymers (Nucleic Acid Sciences),
Vol. 48, 137-153.

Doubly nested pseudoknot: schematic representation



Hilbers et al. 2000

Doubly nested pseudoknot: schematic representation

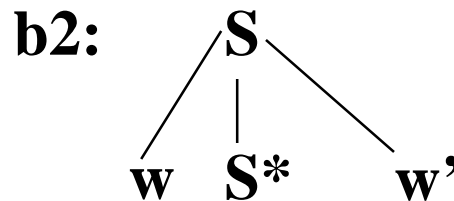
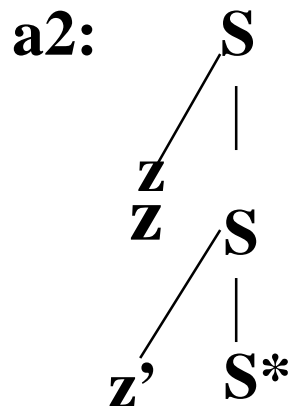
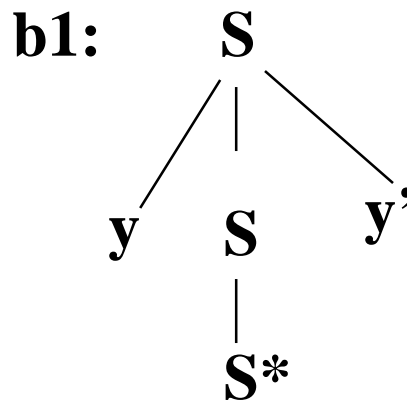
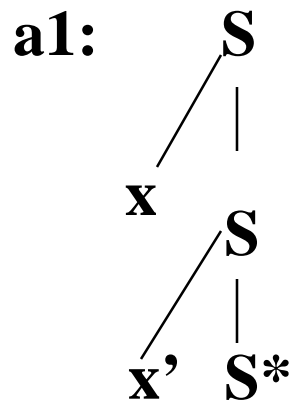


Linear sequence:

X Y Z W Z' X' W' Y'

Doubly nested pseudoknot: TAG grammar

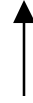
G: Elementary Trees:



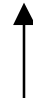
Doubly nested pseudoknot: TAG grammar

Derivation tree:

a1



b1



a2



b2

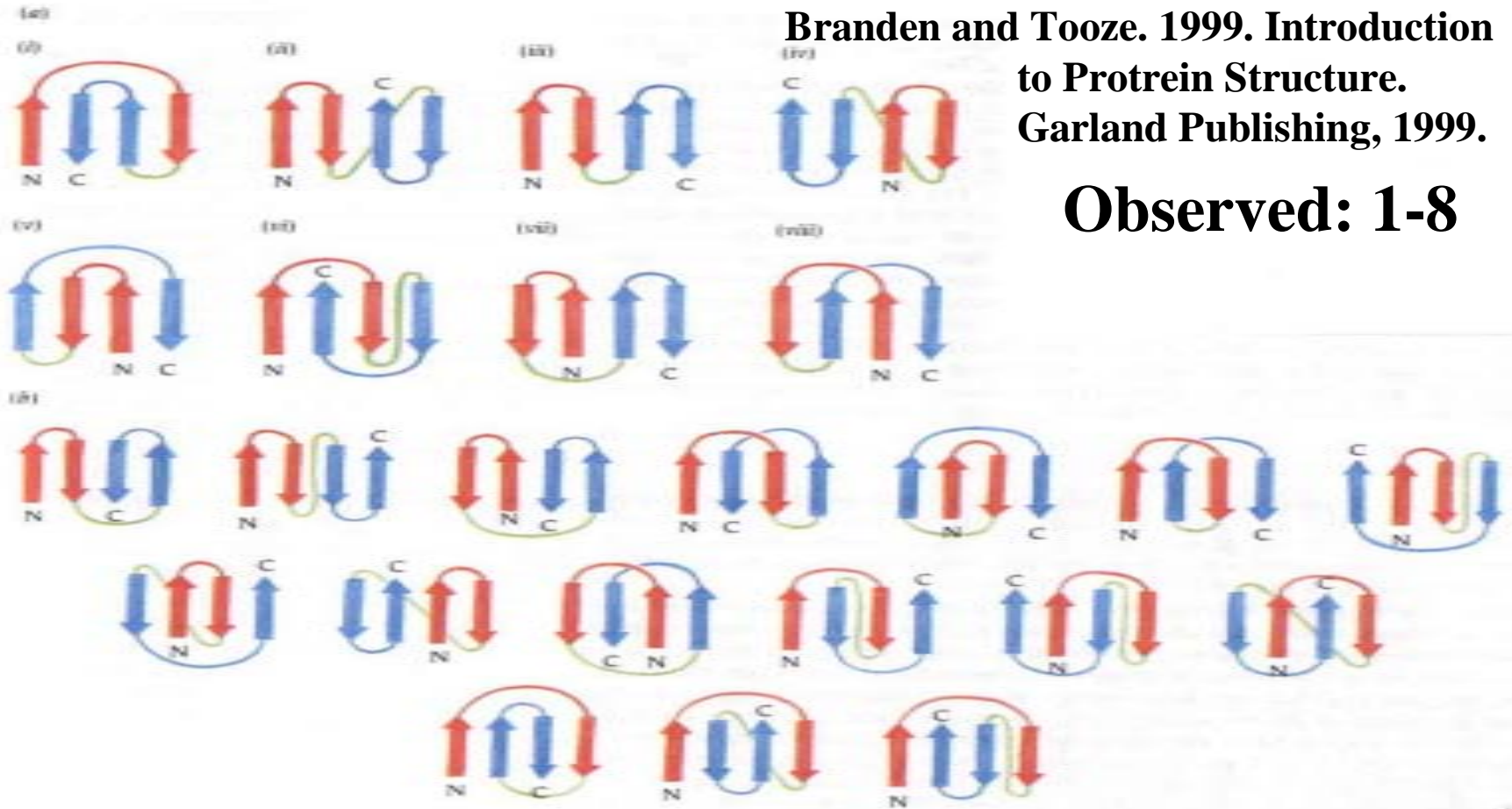
Linear sequence:

X **Y** **Z** **W** **Z'** **X'** **W'** **Y'**

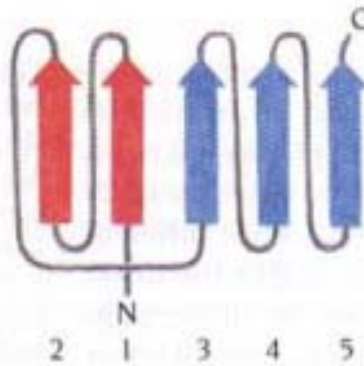
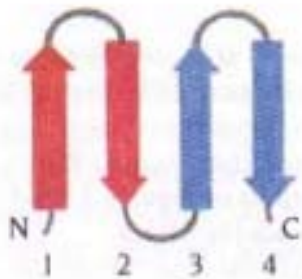
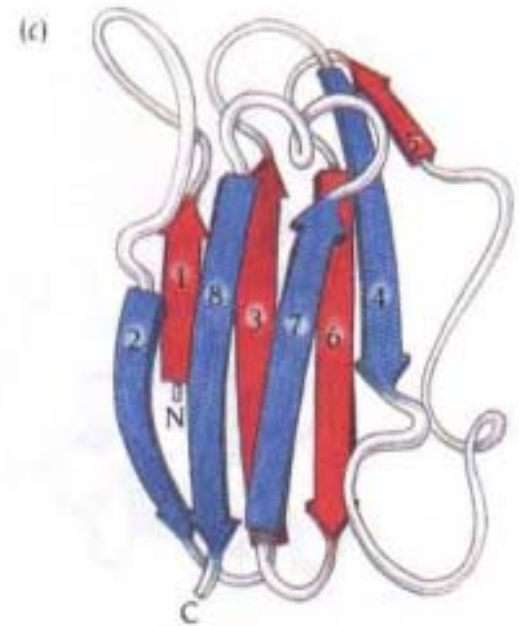
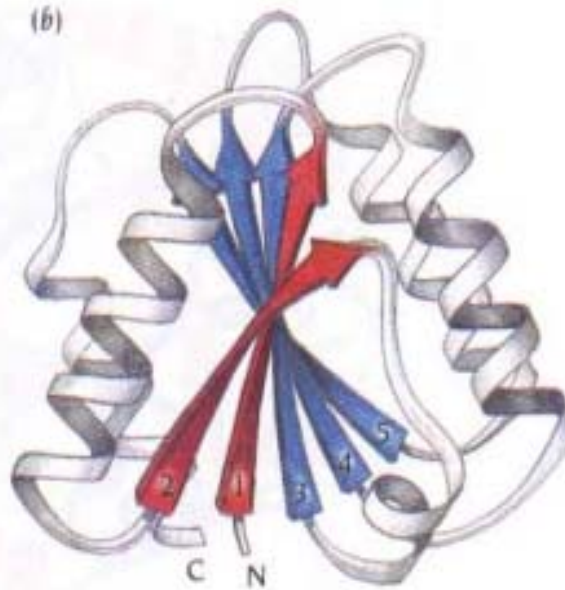
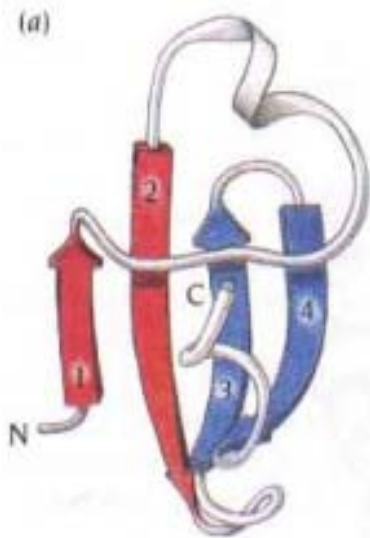
Some structural motifs

- Many complex structures can also be characterized
- These consist of **parallel strands** (crossing dependencies) and **anti-parallel strands** (nested dependencies) connected to each other in various complex ways
- The challenge is to connect this work to the work that deals with the **distribution of energies associated with the different configurations (partition functions)**
 - There are connections of this work to the work in statistical NLP, in particular in parsing

Some structural motifs:



Not observed: 9-24



THANKS!

- Zellig Harris, Lila Gleitman, Naomi Sager, Bruria Kauffman, Carol Chomsky, Phil Hopely
- Leon Levy, Masako Takahashi, Richard Upton, Johnson Hart, T. Yokomori, Lynette Hirschman, K. Vijayshanker, Tony Kroch, David Weir, Yves Schabes, Anne Abeille, Tilman Becker, Robert Frank, Owen Rambow, Mitch Marcus, Mark Steedman, B. Srinivas, Christy Doran, Beth Hockey, Seth Kulick, Anoop Sarkar, Martha Palmer, Maribel Romero, Laura Kallmeyer, Carlos Prolo, Alexandra Kinyon, Libin Shen, David Chiang, Fernando Pereira, Ken Dill
- Ralph Weishedel, Stan Rosenschein, Jerry Kaplan, Eric Mays, Kathy McKeown, Kathy McCoy, Bonnie Webber, Steve Kuhn, Scott Weinstein, Barbara Grosz, Ellen Prince, Marilyn Walker, Eleni Miltsakaki, Kate Forbes, Cassandra Creswell, Rashmi Prasad