Robust parsing and LGSoar

Deryle Lonsdale and Tory Anderson

Different types of trees

Ways of viewing a tree

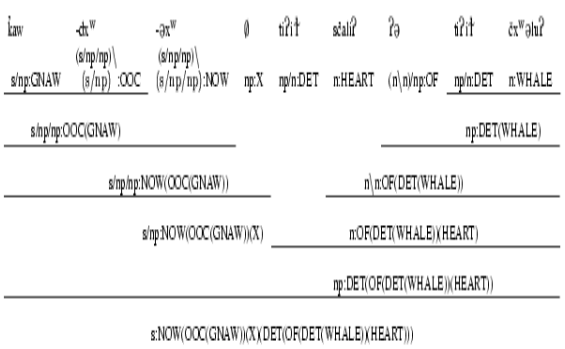
Parsing

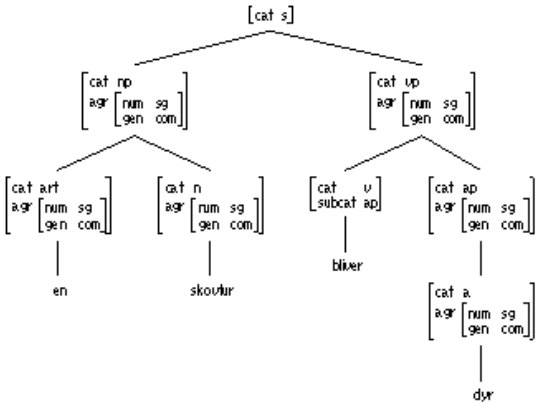
- Start with text (e.g. sentence)
- Label each of the elements (e.g. words)
- Diagram the relationships between elements: parse tree
- Why?
  - Shows constituency
  - Visual representation of content
  - Useful for future reference (e.g. treebanks)

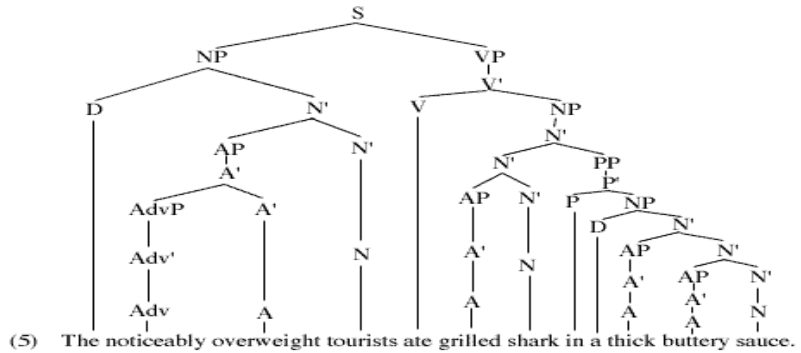
Linguistic theories and parsing

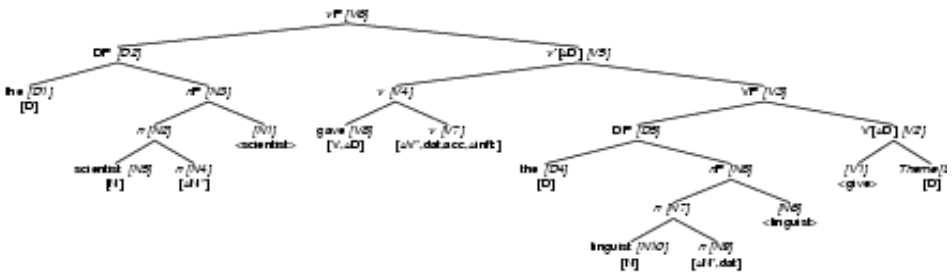
- LFG (KANT)
- GB/P&P (NL-Soar)
- Minimalist Program (XNL-Soar)
- SFG (NIGEL)
- HPSG (Verbmobil)
- Categorial grammar (ALE)
- RST (PENMAN)
- TAG (XTAG)
- STATISTICS (CANDIDE)
- etc. etc.

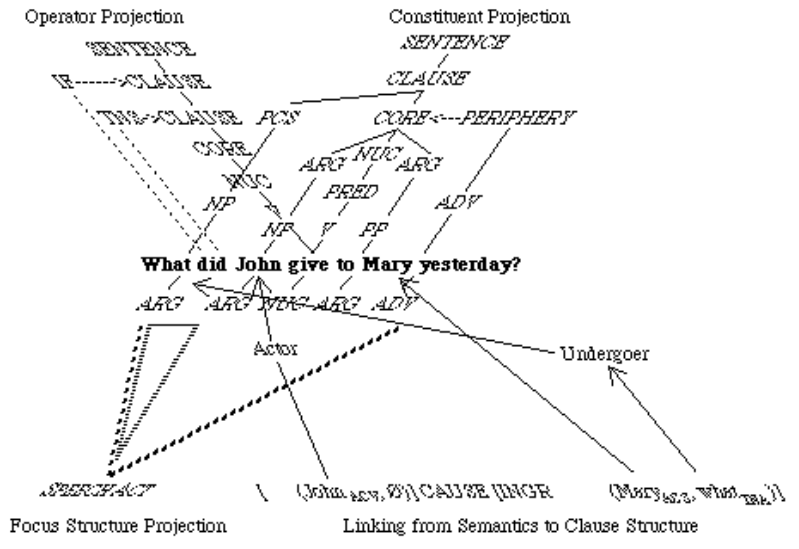
Different kinds of parse trees











Robust parsing

- Most parsers based on linguistic theory
- Linguistic theories assume grammatical input
- Much of language use not entirely grammatical
  - L2 English
  - Spoken language
  - Controlled language / sublanguage
  - Headlines
  - PowerPoint slide bullets
- Traditional parsers don't handle these types of language well

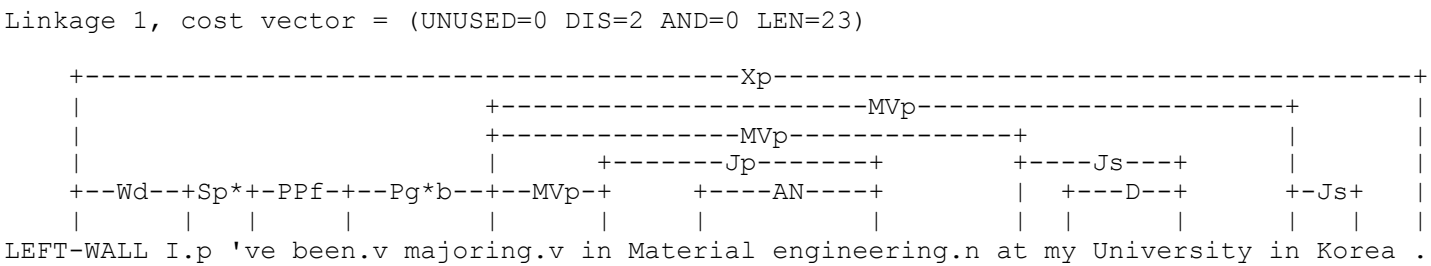
The LG parser

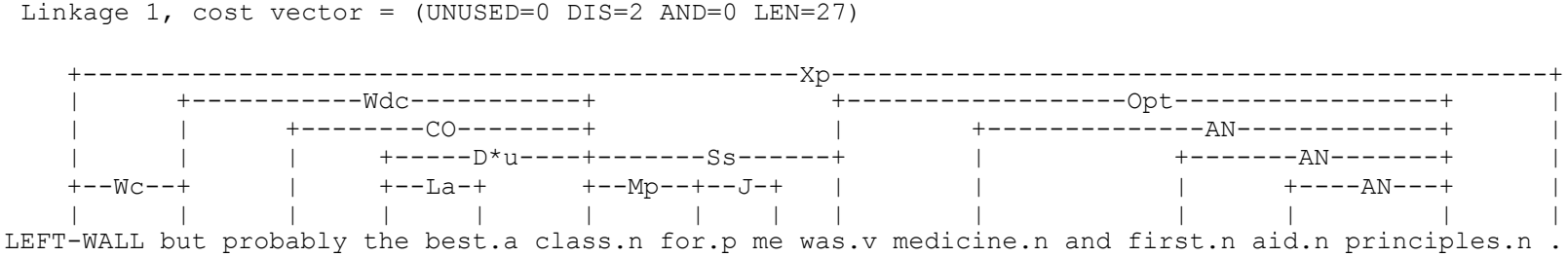
- Freely available for research purposes
- Robust (e.g. information retrieval, MT)
- Calculates simple, explicit relations
- Fast
- Written in C
- Not based on any linguistic theory
- More appropriate for some tasks than traditional phrase-structure parsers

Link Grammar

- What is a link?
  - Shows a relationship between pairs of words
    - Subject + verb
    - Verb + object
    - Preposition + object
    - Adjective + adverbial modifier
    - Auxiliary + main verb
  - Labels each relationship accordingly
- Potential link types are specified by technical rules
- Possible to score linkages, penalize links

LG example parses

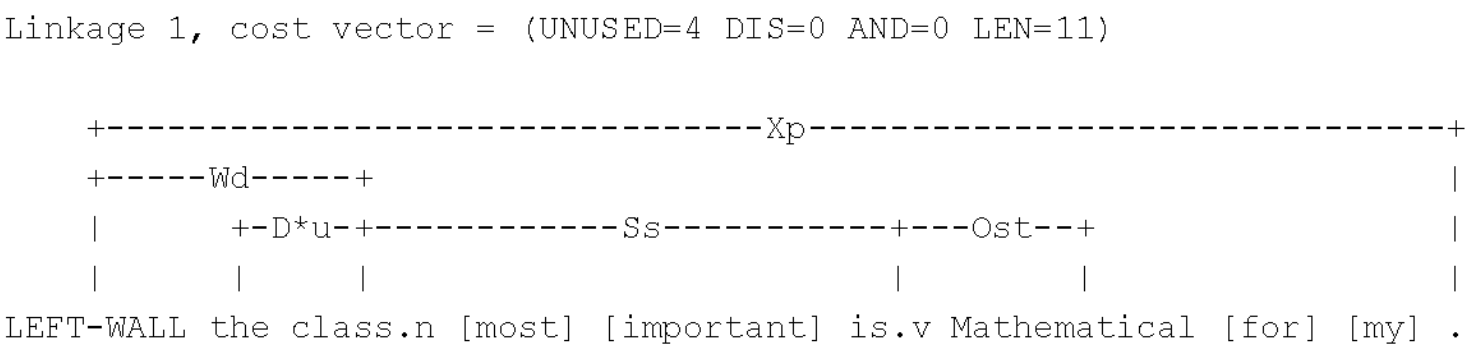




Previous LG applications

- Grading EFL essays
- Extracting biographic facts from genealogical documents
- Analyzing newspaper headlines
- Clinical trial records and patient data matching
- Other languages (Farsi, Arabic, Lushootseed)

Ill-formed sentence



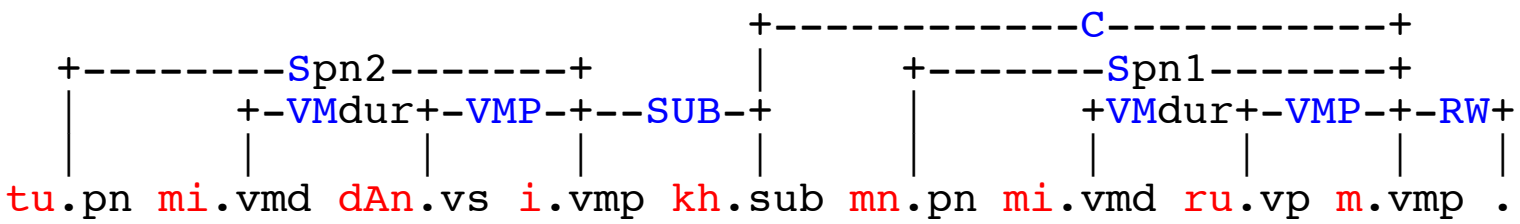
Skipping problematic words

```
Mary married I think, 23 November 1661, Samuel Gay.
No complete linkages found.

          +-------------------------Xc-------------------------+
          +-----------------------Osn-----------------------+  |
          +------------------Xc------------------+          |  |
          +-------------MVp------------+         |          |  |
  +--Ss--+                       +--TM-+--TY--+  |    +--G-+  |
  |      |                       |     |      |  |    |    |  |
Mary married.v [I] [think] [,] 23 November 1661 , Samuel Gay .
```

Parsing Persian with LG

<tu midAni kh mn mirum>
"you know that I am going"



Parsing Lushootseed

```
linkparser> bE+ Lil +t +Eb +ExW ?ElgWE? ?E ti?E? bE+ ?Es+ istE?.

++++Time   0.07 seconds (0.20 total)

Found 2 linkages (2 had no P.P. violations)
  Linkage 1, cost vector = (UNUSED=0 DIS=2 AND=0 LEN=28)

    +-----------------------------Xp-----------------------------+
    |           +-----------EM----------+                        |
    |           +--------PA-------+     +----------P---------+   |
    |           +----ASP----+     |     |   +------DT------+   |
    +----Wd----+--MD--+    |     |     |   |    +----AD---+   |
    |       +-AD+-TX+  |    |     |     |   |    |    +-STV+   |
    |       |   |   |  |    |     |     |   |    |    |    |   |
LEFT-WALL bE+ Lil +t +Eb +ExW ?ElgWE? ?E ti?E? bE+ ?Es+ istE? .
```

Sample link specifications

```
<pref-asp1>: {(PRF- or STV- or PRG-)};
<pref-asp2>: {HAB-} & {DUB-} & {AD-};
<predprefs>: {NZ-} & {<pref-asp1>} & {SX-} & {<pref-asp2>} & {(FUT- or PT-)};

<root-main>:  <predprefs> & {DT-} & {LX+} & {BNF+} & {TX+} & {TC+} & {ACH+} & {TC+} & {TX+} &
    {ASP+};
<main-args>: {P-} & {GEN-} & {WH-} & {SOs+} & {MV+};

<root-ditrx>: <predprefs> & {DT-} & {LX+} & {BNF+} &  TX+  & {TC+} & {ACH+} & {TC+} & {TX+} &
    {ASP+};
<ditrx-args>: {P-} & {GEN-} & {WH-} & {SOs+} & {EX+} & {SOo+} & {MV+};


<root-middle>: <predprefs> & {DT-} & {LX+} & {BNF+} & {TX+} & {TC+} & MD+ & {ACH+} & {TC+} &
    {TX+} & {ASP+};
<middle-args>: {P-} & {GEN-} & {WH-} &  (({PA+} & {EM+}) or ({EM+} & {PA+})) & {MV+};

<pred1>: ((<root-main> & <main-args>) or
      (<root-middle> & <middle-args>) or
          (<root-ditrx> & <ditrx-args>)
     ) & {Wd-};
```

LG-Soar: LG + Soar + DRT

- Parse input via LG parser
- Input words, links into Soar
- Productions to identify and infer:
  - Entities: discourse referents
  - Attributes: properties of entities
  - Actions, states
  - Other relationships: spatial, temporal
  - Anaphor, deixis, other pragmatic content

Discourse Representation Theory

- Discourse Representation Structure
- Discourse referents:
  - Variables, representing objects; anything that can serve as the antecedent for an anaphor
- Conditions:
  - Represent properties and relationships
- Examples:
  - Thomas(u)
  - v married w

Predicate logic equivalent

- Thomas Smith, Haverhill, married at Andover 6 January 1659, Unice Singletary of Salisbury.

```
u v w x m n o y z a
Unice(y), Singletary(z), prep("at", x), verbal("married", v, x)
propername=uv
modifier="Haverhill"
Thomas(u), Smith(v), Haverhill(w), Andover(x), 6(m), January(n),
propername=yz
time(day m, month n), 1659(o), time(month n, year o), Salisbury(a),
modifier="January"
modifier="Andover"
```

LG-Soar current status

- Most complete running system in Soar 7
- Conversion to Soar 9 underway
- Most of code updated to Soar 9 (finally!)
- DRS output format needs updating (currently Tcl)
- Newer version of LG parser released, need to update

Conclusions

COAL

- Not as useful when linguistics required
- No cognitive modeling: Soar is just a programming environment
- Linguistic grammar development very opaque

NUGGETS

- Robust parsing
- Flexible: various applications
- Soar 9.3 underway, largely complete
- Interest growing