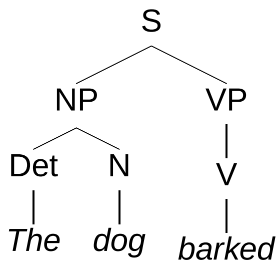

$$\begin{bmatrix} \texttt{LTOP} & h_1 \\ \texttt{INDEX} & e_2 \\ \texttt{RELS} & \left\langle \begin{bmatrix} \textit{prpstn\_m\_rel} & \\ \texttt{LBL} & h_1 \\ \texttt{MARG} & h_3 \end{bmatrix} \begin{bmatrix} \textit{def\_q\_rel} & \\ \texttt{LBL} & h_4 \\ \texttt{ARG0} & x_5 \\ \texttt{RSTR} & h_6 \\ \texttt{BODY} & h_7 \end{bmatrix} \begin{bmatrix} \textit{“dog\_n\_rel”} & \\ \texttt{LBL} & h_8 \\ \texttt{ARG0} & x_5 \end{bmatrix} \begin{bmatrix} \textit{“bark\_v\_rel”} & \\ \texttt{LBL} & h_9 \\ \texttt{ARG0} & e_2 \\ \texttt{ARG1} & x_5 \end{bmatrix} \right\rangle \\ \texttt{HCONS} & \langle h_3 =_q h_9,\ h_6 =_q h_8 \rangle \end{bmatrix}$$

Computational Linguistics (INF2820 — Beyond CFGs)

$$\textit{phrase}\begin{bmatrix} \texttt{HEAD} & \boxed{1} \\ \texttt{SPR} & \langle\rangle \\ \texttt{COMPS} & \boxed{3} \end{bmatrix} \longrightarrow \boxed{2}\ \textit{phrase}\begin{bmatrix} \texttt{SPR} & \langle\rangle \\ \texttt{COMPS} & \langle\rangle \end{bmatrix},\ \textit{phrase}\begin{bmatrix} \texttt{HEAD} & \boxed{1} \\ \texttt{SPR} & \langle \boxed{2} \rangle \\ \texttt{COMPS} & \boxed{3} \end{bmatrix}$$

Stephan Oepen

Universitetet i Oslo & CSLI Stanford

`oe@ifi.uio.no`

The CKY (Cocke, Kasami, & Younger) Algorithm

for $(0 \leq i < |\textit{input}|)$ do
  $\textit{chart}_{[i,i+1]} \leftarrow \{\alpha \mid \alpha \rightarrow \textit{input}_i \in P\}$;
for $(1 \leq l < |\textit{input}|)$ do
  for $(0 \leq i < |\textit{input}| - l)$ do
    for $(1 \leq j \leq l)$ do
      if $(\alpha \rightarrow \beta_1\,\beta_2 \in P \wedge \beta_1 \in \textit{chart}_{[i,i+j]} \wedge \beta_2 \in \textit{chart}_{[i+j,i+l+1]})$ then
        $\textit{chart}_{[i,i+l+1]} \leftarrow \textit{chart}_{[i,i+l+1]} \cup \{\alpha\}$;

Kim adored snow in Oslo

$[0,2] \leftarrow [0,1] + [1,2]$

$\ldots$

$[0,5] \leftarrow [0,1] + [1,5]$

$[0,5] \leftarrow [0,2] + [2,5]$

$[0,5] \leftarrow [0,3] + [3,5]$

$[0,5] \leftarrow [0,4] + [4,5]$

	1	2	3	4	5
0	NP		S		S
1		V	VP		VP
2			NP		NP
3				P	PP
4					NP

Chart Parsing — Specialized Dynamic Programming

Basic Notions

- Use *chart* to record partial analyses, indexing them by string positions;
- count inter-word vertices; CKY: chart row is *start*, column *end* vertex;
- treat multiple ways of deriving the same category for some substring as *equivalent*; pursue only once when combining with other constituents.

Key Benefits

- Dynamic programming (memoization): avoid recomputation of results;
- efficient indexing of constituents: no search by start or end positions;
- compute *parse forest* with exponential 'extension' in *polynomial* time.

Limitations of the CKY Algorithm

Built-In Assumptions

- *Chomsky Normal Form* grammars: $\alpha \rightarrow \beta_1\beta_2$ or $\alpha \rightarrow \gamma$ ($\beta_i \in C$, $\gamma \in \Sigma$);
- breadth-first (aka exhaustive): always compute all values for each cell;
- rigid control structure: bottom-up, left-to-right (one diagonal at a time).

Generalized Chart Parsing

- Liberate order of computation: no assumptions about earlier results;
- *active edges* encode partial rule instantiations, 'waiting' for additional (adjacent and passive) constituents to complete: $[1, 2, \text{VP} \rightarrow \text{V} \bullet \text{NP}]$;
- parser can fill in chart cells in *any* order and guarantee completeness.

Generalized Chart Parsing

- The parse *chart* is a two-dimensional matrix of *edges* (aka chart items);
- an edge is a (possibly partial) rule instantiation over a substring of input;
- the chart indexes edges by start and end string position (aka vertices);
- dot in rule RHS indicates degree of completion: $\alpha \rightarrow \beta_1 ... \beta_{i-1} \bullet \beta_i ... \beta_n$
- *active* edges (aka *incomplete* items) — partial RHS: $[1, 2, \text{VP} \rightarrow \text{V} \bullet \text{NP}]$;
- *passive* edges (aka *complete* items) — full RHS: $[1, 3, \text{VP} \rightarrow \text{V NP}\bullet]$;

The Fundamental Rule

$$[i,\ j,\ \alpha \rightarrow \beta_1 ... \beta_{i-1} \bullet \beta_i ... \beta_n] + [j,\ k,\ \beta_i \rightarrow \gamma^{+} \bullet]$$
$$\mapsto [i,\ k,\ \alpha \rightarrow \beta_1 ... \beta_i \bullet \beta_{i+1} ... \beta_n]$$

An Example of a (Near-)Complete Chart

	1	2	3	4	5
0	NP → NP • PP S → NP • VP NP → kim •				S → NP VP •
1		VP → V • NP V → adored •	VP → VP • PP VP → V NP •		VP → VP • PP VP → VP PP • VP → V PP •
2			NP → NP • PP NP → snow •		NP → NP • PP NP → NP PP •
3				PP → P • NP P → in •	PP → P NP •
4					NP → NP • PP NP → oslo •

$_0$ *Kim* $_1$ *adored* $_2$ *snow* $_3$ *in* $_4$ *Oslo* $_5$

(Even) More Active Edges

	0	1	2	3
0	S → • NP VP NP → • NP PP NP → • kim	S → NP • VP NP → NP • PP NP → kim •		S → NP VP •
1		VP → • VP PP VP → • V NP V → • adored	VP → V • NP V → adored •	VP → VP • PP VP → V NP •
2			NP → • NP PP NP → • snow	NP → NP • PP NP → snow •
3				

- Include all grammar rules as *epsilon* edges in each *chart*$_{[i,i]}$ cell.
- after initialization, apply *fundamental rule* until fixpoint is reached.

Our ToDo List: Keeping Track of Remaining Work

The Abstract Goal

- Any chart parsing algorithm needs to check all pairs of adjacent edges.

A Naïve Strategy

- Keep iterating through the complete chart, combining all possible pairs, until no additional edges can be derived (i.e. the fixpoint is reached);
- frequent attempts to combine pairs multiple times: deriving ‘duplicates’.

An Agenda-Driven Strategy

- Combine each pair exactly once, viz. when both elements are available;
- maintain *agenda* of new edges, yet to be checked against chart edges;
- new edges go into agenda first, add to chart upon retrieval from agenda.

Recap: Grammatical Categories

Number — Person — Case — Gender

That dog barks. — Those dogs bark.

I bark. — You bark. — They bark. — Sam shaved himself.

We bark. — You bark. — Those dogs bark.

I saw her. — She saw me. — My dog barked.

...

Tense — Aspect — Mood

The dog barks. — The dog barked — The dog will bark.

The dog has barked. — The dog is barking.

If I were a carpenter, ...

...

Limitations of (Our) Context-Free Grammars

Agreement and Valency (For Example)

That dog barks.

**That dogs barks.*

**Those dogs barks.*

The dog chased a cat.

**The dog barked a cat.*

**The dog chased.*

**The dog chased a cat my neighbours.*

The cat was chased by a dog.

**The cat was chased of a dog.*

...

Agreement and Valency in Context-Free Grammars

A Really Complicated Language

[...] *omdat* *ik* *Henk* *de* *nijlpaarden* *zag* *voeren* *.*

A Really Complicated Language

[...] *omdat ik Jan Henk de nijlpaarden zag helpen voeren .*

More Terminology: Grammatical Functions

Licensing — Government — Agreement

The dog barks. — **The dog a cat barks* — **The dog barks a cat.*

Kim depends on Sandy — **Kim depends in Sandy*

The class meets on Thursday in 3B at 12:15.

- **Constituent** node in analysis tree (terminal or instantiation of rule);
- **Head** licenses additional constituents and can govern their form;
- **Specifier** precedes head, singleton, nominative case, agreement;
- **Complement** post-head, licensed and governed, order constraints;
- **Adjunct** 'free' modifier, optional, may iterate, designated position;
- **Government** directed: a property of c_1 determines the form of c_2;
- **Agreement** bi-directional: co-occurence of properties on c_1 and c_2.