Machine learning and the cognitive science of natural language

Alexander Clark

Department of Computer Science
Royal Holloway, University of London

Three fields

Fields

- Linguistics
- Cognitive Science
- Machine learning

- Linguistics and cognitive science
- Computational Linguistics and Machine Learning

All three

Computationally explicit and cognitively plausible models of language acquisition.

Outline

Cognitive science and linguistics

Jerry Fodor:

> *The Argument from the Poverty of the Stimulus is the existence proof for the possibility of cognitive science.*

Cognitive science and linguistics

Jerry Fodor:

> *The Argument from the Poverty of the Stimulus is the existence proof for the possibility of cognitive science.*

Chomsky's innovations

- Inadequacy of simple behaviorist models (c.f. Skinner)
- Need to hypothesize a very rich internal structure to account for the complexities of language
- Computationally (formally) explicit models of natural language – the Chomsky hierarchy.
- View of linguistics as a branch of psychology

Unsupervised Learning

Fundamental problem of linguistics

Chomsky's questions (1986)

1. What constitutes knowledge of a language?
2. How is this knowledge acquired by its speakers?

Jackendoff (2008)

1. Descriptive constraint: the class of languages must be sufficiently rich to represent natural languages
2. Learnability constraint: there must be a way for the child to learn these representations from the data available
3. Evolutionary constraint: it must not posit a rich, evolutionarily implausible language faculty

Outline

The argument from the poverty of the stimulus

Many different versions:
Hornstein and Lightfoot

> *People attain knowledge of the structure of their language for which no evidence is available in the data to which they are exposed as children.*

Pullum and Scholz (2002)

Children learn natural languages:

- Rapidly
- Consistently
- Without explicit instruction
- The data is sparse, incomplete and noisy
- Uniform outcome

This cannot be accounted for by standard models of learning.

Empirical version

Perfors et al. (2006)

1. The student is hungry
2. Is the student hungry?

Empirical version

Perfors et al. (2006)

1. The student is hungry
2. Is the student hungry?
3. The student who is in the garden is hungry.

Empirical version

Perfors et al. (2006)

1. The student is hungry
2. Is the student hungry?
3. The student who is in the garden is hungry.
4. Is the student who is in the garden hungry?
5. *Is the student who in the garden is hungry?

Children do not see examples of type 4, but produce the right examples.

There is a factual problem.

The argument from the poverty of the stimulus

Chomsky (1965), pp. 57-58

"In brief, it seems clear that the present situation with regard to the study of language learning is essentially as follows. We have a certain amount of evidence about the character of the generative grammars that must be the "output" of an acquisition model for language. This evidence shows clearly that taxonomic views of linguistic structure are inadequate and that knowledge of grammatical structure cannot arise by application of step-by-step inductive operations (segmentation, classification, substitution procedures, filling of slots in frames, association, etc.) of any sort that have been developed within linguistics, psychology, or philosophy. Further empiricist speculations contribute nothing that even faintly suggests a way of overcoming the intrinsic limitations of the methods that have so far been proposed and elaborated."

Linguistic nativism
The Language Instinct

Definition

Linguistic nativism is the claim that language aquisition proceeds largely through innate, language-specific mechanisms and representations.
A lot of grammar is encoded in the genome.

Vacuous claim

Clearly we have some innate ability to acquire language: since we do and lobsters don't.
The debate is whether it is domain-specific or not.

Linguistic nativism
The Language Instinct

Definition

Linguistic nativism is the claim that language aquisition proceeds largely through innate, language-specific mechanisms and representations.
A lot of grammar is encoded in the genome.

Vacuous claim

Clearly we have some innate ability to acquire language: since we do and lobsters don't.
The debate is whether it is domain-specific or not.

Chomsky seems no longer to subscribe to this view.

The formal version of the APS

Ken Wexler:

> *The strongest most central arguments for innateness thus continue to be the arguments from APS and learnability theory. . . . The basic results of the field include the demonstration that without serious constraints on the nature of human grammar, no possible learning mechanism can in fact learn the class of human grammars.*

- The formal arguments do not support linguistic nativism, even under the most optimistic interpretations.

(Clark and Lappin, 2010)

Problem with the argument

Key distinction:

- Hypothesis class of a learning algorithm
- The class of languages that the algorithm can learn

One can show that the learnable class is restricted in some way, but you can't show that the former needs to be restricted.

Example

Under distribution-free uniform PAC-learning

- The learnable class must have finite VC-dimension
- The hypothesis class does not need to be bounded (Haussler and Kearns 1991)

Study of Language acquisition

Pinker (1990)

To understand how X is learned, you first have to understand what X is.

Study of Language acquisition

Pinker (1990)

To understand how X is learned, you first have to understand what X is.

Crain and Pietroski (2001)

First, one tries to find principles that characterize human grammars; *then* one tries to determine which aspects of these grammars could plausibly be learned from experience, and which are more likely to be innately specified.

Standard methodology

- Step 1: Construct a descriptively adequate representation
- Step 2: Try to design learning algorithms for those representations

Step 1
Construct a descriptively adequate grammar

This failed

- No-one ever managed to make a descriptively adequate grammar for any natural language, not even English.
- In order to account for new facts (e.g. Swiss German) representations were made more powerful and expressive.
- Statistical parsers do not separate grammatical from ungrammatical sentences (Okanohara and Tsujii, 2007; Berwick and Fong, 2008)
- Generative grammarians have largely abandoned the task of constructing large scale grammars.

Step 2
Come up with a learning algorithm

This also failed.

- Learning even regular grammars is computationally hard: Angluin and Kharitonov (1995)
- We have some heuristic algorithms that can induce crude constituent structure (Klein and Manning, 2004)
- The classes of representations we need have even richer, deeper and more abstract hidden structure: (LTAG, $ACG_{2,4}$, ...)
- It is out of the question to construct learning algorithms for these classes.

Tension

Chomsky, 1986

To achieve descriptive adequacy it often seems necessary to enrich the system of available devices, whereas to solve our case of Plato's problem we must restrict the system of available devices so that only a few languages or just one are determined by the given data. It is the tension between these two tasks that makes the field an interesting one, in my view.

Principles and Parameters models

Principles and Parameters

Language is entirely innately specified apart from a finite number of binary valued parameters

- Evolutionarily implausible
- No good learning model
- No agreement on parameters
- No tension
- Currently being abandoned.

Linguists don't know what the representations are

A Cambridge quote

"At the most fundamental level, it is not clear that there is any meaningful empirical motivation for the representational assumptions of any current formal model of syntax."
(Blevins, J., 2009)

Linguists cannot agree whether the head of "the cat" is "the" or "cat". Nor can they produce any empirical evidence to decide between the two.
(Matthews, P.; 2007)

Linguists don't know what the representations are

A Cambridge quote

"At the most fundamental level, it is not clear that there is any meaningful empirical motivation for the representational assumptions of any current formal model of syntax."
(Blevins, J., 2009)

Linguists cannot agree whether the head of "the cat" is "the" or "cat". Nor can they produce any empirical evidence to decide between the two.
(Matthews, P.; 2007)

- We don't know what the representations are but we do know that they are learnable!

Reasonable Research Strategy

Slogan

Put learnability first!

- If you construct a super-powerful class of languages with no thought of learnability, you won't be able to learn them.
- Rather, design representations from the ground up to be learnable.

Reasonable Research Strategy

Slogan

Put learnability first!

- If you construct a super-powerful class of languages with no thought of learnability, you won't be able to learn them.
- Rather, design representations from the ground up to be learnable.

Strategy

- Step 1: build simple learnable representations
- Step 2: gradually try to increase their expressive power, while maintaining learnability

Outline

Current NLP

Prehistory: manually constructed programs for a specific task.

Current solution

Use supervised learning from annotated data.

This has problems:

- It is a poor solution from an engineering point of view: knowledge engineering bottleneck
- It is no solution to the cognitive science problem.

POS tagging

In some languages the lexical class of a word is not determined by its surface form.

Task

Given a sequence of words $x_1, x_2, \ldots x_n$

- He rose dripping from the lake
- He handed her the rose

Assign to each a POS tag: N, V, etc.

$y_1, \ldots, y_n$

Standard approach

Ken Church's HMM tagger:

- Joint probabilistic model $p(x_1 \ldots x_n, y_1, \ldots y_n)$
- HMM with states identified with POS tags y_i
- ML training – Viterbi decoding

Supervised Parsing

Task

Given a sequence of words $x_1, \ldots x_n$
Assign a latent tree constituent structure tree.
Data is a treebank

Geoff Sampson (1986) – APRIL

- Stochastic model from SUSANNE corpus
- Simulated annealing parser.

Charniak, Collins etc.

Example of plausible supervision

Sometimes it is cognitively plausible to have a supervised learning:

Stress assignment

- Lexical specification: récord/recórd
- Phonologically specified: always on the penultimate syllable

Supervised learning problem: given unknown word where is the stress?

Morphological

English past tense: “the fruit fly of linguistics” Pinker

- walk/walked, go/went, break/broke
- plausible that the learner can identify pairs
- Supervised learning problem – learn transduction $\Sigma^* \rightarrow \Sigma^*$

Outline

Unsupervised learning

Two motivations in CL –

- Annotation bottleneck
  - Resource poor languages
  - Huge amounts of data
- Cognitive modelling

Tasks

- POS induction (Clark, 2003)
- Segmentation – NPB (Goldwater, Johnson et al.)
- Morphology learning (Goldsmith, 2001)
- Grammar induction

Grammar induction

This is the central problem:

Regular languages

Other problems are largely modelled by regular or finite state models:
we know (more or less) how to learn finite state models

- Empirical approaches – implement algorithms
- Theoretical approaches – grammatical inference

Empirical results

Standard assumptions in unsupervised learning in NLP

- Use real data: positive only,
- Evaluate against gold standard – linguistic annotations
- Use heuristic algorithms.

Klein and Manning, 2002

- WSJ10 – sentences of length < 10 from WSJ
- Evaluate using modified PARSEVAL metrics
- Binary tree branching constraint with distributional heuristic.

Two problems of grammar induction

Information theoretic problems

- Absence of negative data (Gold, 1967)
- VC-dimension (Vapnik, 1998)
- Sparsity, Noise etc.

We know how to attack these problems: MDL, NPB
Not specific to language

Two problems of grammar induction

Computational problems

Complexity of finding the best hypothesis

- Gold (1978), Kearns and Valiant (1989) ...
- Specific to certain classes of representation
- Often based on embedding cryptographic problems in learning problems

Tractable Cognition Thesis (van Rooij, 2008)

Human cognitive capacities are constrained by the fact that humans are finite systems with limited resources for computation.

Too hard to try to solve both of these problems together. Here we try to solve the second and assume the first has been dealt with.

Overview

	Inefficient	Efficient
Positive data and MQs	Gold (1967)	?
Stochastic data	Horning (1969)	?
	Angluin (1988)	
	Chater and Vitanyi (2007)	

Regular inference
A success story

Paradigm	Learnable class	
Positive Data	reversible languages	Angluin (1982)
Queries	regular languages	Angluin (1987)
Positive and Negative	regular languages	Oncina and Garcia (1992)
Stochastic data	acyclic PDFAs	Ron et al (1994),
	regular languages	Carrasco and Oncina (199
	regular languages	Clark and Thollard (2004)

> *These results suggest the presence of probabilistic data largely compensate for the absence of negative data. (Angluin, 1988)*

We will assume positive data and membership queries as a place holder for more realistic models. (Clark and Lappin, 2009)

Why are DFAs learnable?

All of these models learn the minimal DFA:

Residual languages

$u^{-1}L = \{v | uv \in L\}$
right congruence classes

States

$L(q) = \{u | \delta(q, u) \in Q_F\}$
strings generated from a state

The minimal DFA has states which exactly correspond to residual languages.
It is *objective*.

Empiricist models

Slogan

The structure of the representation should be based on the structure of the language, not something arbitrarily imposed on it from outside.

- Identify some structure in the language
- Show how that structure can be observed
- Construct a representation based on that structure
- Richer structures will give you more powerful representations

Go backwards

Normal direction

Function from representation to language
Context free grammar $G \rightarrow$ context free language $L(G)$
Non-terminal $\rightarrow$ set of strings derived from non-terminal

Go backwards

Normal direction

Function from representation to language
Context free grammar $G \rightarrow$ context free language $L(G)$
Non-terminal $\rightarrow$ set of strings derived from non-terminal

Opposite Direction

Function from language to representation
$L \rightarrow R(L)$
From set of strings $\rightarrow$ representational primitive of formalism

Ideally $L(R(L)) = L$.

Outline

Summary

Technical Content

Distributional lattice grammars

- Richly structured context sensitive representation; Class of languages seems a good match to the class of natural languages;
- Efficient, correct algorithms for learning based on distributional learning;
- Solid theoretical foundation in the theory of residuated lattices
- Formal results use a symbolic learning paradigm

Where's the data?

- We can use data to distinguish between competing theories.
- There are no satisfactory theories at the moment
- Two candidates:
  1. Construction grammar (Tomasello, Goldberg . . .)
  2. Principles and Parameters (Chomsky, 1981, Yang 2002 . . .)

Examples and proof

Mathematical proof should be more convincing.
Examples are to illuminate rather than convince

Basic assumptions

A finite set of symbols:
$\Sigma = \{\mathit{the}, \mathit{a}, \mathit{cat}, \mathit{dog}, \mathit{is}, \ldots, \mathit{assumption}, \ldots\}$
Σ^* is the set of all finite strings.
L is the subset of grammatical sentences $\{$ the cat is dead, I ran away $\ldots\}$
$\Sigma^* \setminus L = \{$ the the, cat pattern helicopter, $\ldots\}$
Very crude: better to have a distribution over Σ^*

Example

$\Sigma = \{a, b\}$
$L = \{a^n b^n | n \geq 0\}$
$L = \{\lambda, ab, aabb, \ldots\}$
$\Sigma^* \setminus L = \{a, b, ba, bb, \ldots aabbb, aaba \ldots\}$

Learning problem

Given some information about L construct a representation G such that G defines the language L.
Typically:

- Sequence of examples $w_1, w_2 \ldots$
- Only constraint: $\{w_i\} = L$
- Very weak constraint: we compensate by allowing learner to query whether w is in L
- We require convergence to a right answer

Very limited source of information – no context, no semantics.

Distributional learning

Zellig Harris

Default assumption

Generalise in some way from a set of examples.

Natural algorithmic idea:

- Look at the doggy
- Look at the car
- Look at the biscuit
- Look at the blue car
- the doggy is over there
- the biscuit is over there
- ...

Question: what classes of languages can be learned using this approach?

Problems

A classic example from Chomsky:

- John is easy to please
- John is eager to please

Problems

A classic example from Chomsky:

- John is easy to please
- John is eager to please
- They are ready to eat

Problems

A classic example from Chomsky:

- John is easy to please
- John is eager to please
- They are ready to eat

Displaced constituents

- This is the book that John said that Mary had ...

Distribution

Example

That man over there is bothering me

Distribution

Example

That man over there is bothering me

Split

- A substring “man over”
- A context “That _ there is bothering me”

This is “observable”

Distribution

Classic idea from structuralist linguistics:

Context (or *environment*)

A context is just a pair of strings $(l, r) \in \Sigma^* \times \Sigma^*$.
$(l, r) \odot u = lur$
$f = (l, r)$.
Special context (λ, λ)

Given a language $L \subseteq \Sigma^*$.

Distribution of a string

$C_L(u) = \{(l, r) | lur \in L\} = \{f | f \odot u \in L\}$

"Distributional Learning" models/exploits the distribution of strings;

Example

$L = \{a^n b^n | n \geq 0\}$

$C_L(a) = \{(\lambda, b), (a, bb), (a, abbb) \dots\}$
$C_L(aab) = \{(\lambda, b), (a, bb), \dots\}$
$C_L(aaabb) = \{(\lambda, b), (a, bb), \dots\}$

English

$C_L(cat) = \{$ look at the _, the _ is on the mat $\dots\}$
$C_L(dog) = \{$ look at the _, the _ is on the mat $\dots\}$

Distributional learning

Several reasons to take distributional learning seriously:

- Cognitively plausible (Safran et al)
- It works in practice: large scale lexical induction (Curran, J. 2003)
- Linguists use it as a constituent structure test (Carnie, A, 2008)
- Historically, PSGs were intended to be the output from distributional learning algorithms.

Chomsky (1968/2006)

“The concept of "phrase structure grammar" was explicitly designed to express the richest system that could reasonable be expected to result from the application of Harris-type procedures to a corpus.”

Distributional learning

Several reasons to take distributional learning seriously:

- Cognitively plausible (Safran et al)
- It works in practice: large scale lexical induction (Curran, J. 2003)
- Linguists use it as a constituent structure test (Carnie, A, 2008)
- Historically, PSGs were intended to be the output from distributional learning algorithms.

Chomsky (1968/2006)

“The concept of "phrase structure grammar" was explicitly designed to express the richest system that could reasonable be expected to result from the application of Harris-type procedures to a corpus.”

Distributional learning

- Try to predict $C_L(u)$
- Learn some finite representation G that defines
  $\phi_G : u \mapsto C_L(u)$
  $\phi_G : \Sigma^* \to 2^{\Sigma^* \times \Sigma^*}$
- $(\lambda, \lambda) \in C_L(u)$ iff $u \in L$

Two problems

- $C_L(u)$ will normally be infinite; so we need some representation $\mathcal{Y}$
- There are an infinite number of strings u in Σ^*; so we need some way of computing $\phi(uv)$ from $\phi(u)$ and $\phi(v)$.

Finite representation

Set of contexts

Take a finite set of contexts F
Including at least (λ, λ)

Finite set of substrings

Take a finite set of substrings K
Including at least λ and Σ
i.e. All $|w| \leq 1$

Any language

L is an arbitrary subset of Σ^*
Data we need is $L \cap (F \odot KK)$
$F \odot KK = \{luvr | (l, r) \in F, u, v \in K\}$

Partition

Congruence classes

We can partition the strings into
$[u] = \{v | v \equiv_L u\}$

Example: $L = \{a^n b^n | n \geq 0\}$

- $[a] = \{a\}$
- $[ab] = \{ab, aabb, \dots\}$
- $[aab] = \{aab, aaabb, \dots\}$
- $\dots$
- $[\lambda] = \{\lambda\}$
- $[ba] = \{ba, bba, \dots\}$

Context free grammar

Suppose we have a grammar and $L(N)$ is the set of strings generated by non-terminal N.

- We have a rule $N \rightarrow PQ$
- This means that $L(N) \supseteq L(P)L(Q)$.

Context free grammar

Suppose we have a grammar and $L(N)$ is the set of strings generated by non-terminal N.

- We have a rule $N \rightarrow PQ$
- This means that $L(N) \supseteq L(P)L(Q)$.

Backwards

Objectively define a collection of sets of strings X, Y, Z
Suppose $X \supseteq YZ$
Then we add a rule $X \rightarrow YZ$.

Example

Congruence classes have nice properties!

$[u][v] \subseteq [uv]$
$[uv] \rightarrow [u][v]$

Problem

Hard to tell whether $u \in [v]$

Example

$L = \{a^n b^n | n \geq 0\}$

$[a] = \{a\}, [abb] = \{abb, aabbb, \dots\}$

$[a][aab] = \{aabb, aaabbb \dots\} \subseteq [aabb] = [ab]$

Example

$L = \{a^n b^n | n \geq 0\}$

$[a] = \{a\}, [abb] = \{abb, aabbb, \dots\}$
$[a][aab] = \{aabb, aaabbb \dots\} \subseteq [aabb] = [ab]$

Grammar

- $S \rightarrow [ab], S \rightarrow [\lambda]$

Example

$L = \{a^n b^n | n \geq 0\}$

$[a] = \{a\}, [abb] = \{abb, aabbb, \dots\}$
$[a][aab] = \{aabb, aaabbb \dots\} \subseteq [aabb] = [ab]$

Grammar

- $S \rightarrow [ab], S \rightarrow [\lambda]$
- $[a] \rightarrow a, [b] \rightarrow b, [\lambda] \rightarrow \lambda$

Example

$L = \{a^n b^n | n \geq 0\}$

$[a] = \{a\}, [abb] = \{abb, aabbb, \dots\}$
$[a][aab] = \{aabb, aaabbb \dots\} \subseteq [aabb] = [ab]$

Grammar

- $S \rightarrow [ab], S \rightarrow [\lambda]$
- $[a] \rightarrow a, [b] \rightarrow b, [\lambda] \rightarrow \lambda$
- $[ab] \rightarrow [aab][b], [ab] \rightarrow [a][b], [ab] \rightarrow [a][abb]$
- $[aab] \rightarrow [a][ab], [abb] \rightarrow [ab][b]$

Example

$L = \{a^n b^n | n \geq 0\}$

$[a] = \{a\}, [abb] = \{abb, aabbb, \dots\}$
$[a][aab] = \{aabb, aaabbb \dots\} \subseteq [aabb] = [ab]$

Grammar

- $S \rightarrow [ab], S \rightarrow [\lambda]$
- $[a] \rightarrow a, [b] \rightarrow b, [\lambda] \rightarrow \lambda$
- $[ab] \rightarrow [aab][b], [ab] \rightarrow [a][b], [ab] \rightarrow [a][abb]$
- $[aab] \rightarrow [a][ab], [abb] \rightarrow [ab][b]$
- Plus $[ba] \rightarrow [b][ba] \dots$
- Plus $[a] \rightarrow [\lambda][a] \dots$

Two Distributional Strategies

Strings

$[u] = \{v | v \equiv_L u\}$
Congruence classes: these are the smallest possible sets

Contexts

$C[l, r] = \{v | lvr \in L\}$
These are the largest possible sets.

(If the languages are substitutable then they coincide)

Old concept

Myhill, 1950

I shall call a system *regular* if the following holds for all expressions μ, ν and all wffs ϕ, ψ each of which contains an occurrence of ν: If the result of writing μ for some occurrence of ν in ϕ is a wff, so is the result of writing μ for any occurrence of ν in ψ. Nearly all formal systems so far cnstructed are regular; ordinary word-languages are conspicuously not so.

Clark and Eyraud, 2005

A language is *substitutable* if $lur, lvr, l'ur' \in L$ means that $l'vr' \in L$.

Why the delay?

Congruence class results

Positive data alone

$lur \in L$ and $lvr \in L$ implies $u \equiv_L v$

Polynomial result from positive data. (Clark and Eyraud, 2005)

k-l substitutable languages, Yoshinaka (2008)

Stochastic data

If data is generated from a PCFG

PAC-learn unambiguous NTS languages, Clark (2006)

Membership queries

An efficient query-learning result

Pick a finite set of contexts F

Test if $C_L(u) \cap F = C_L(v) \cap F$

Limitations

One symbol per congruence class just won't work for natural languages:

- Congruence classes are very many and very close together
- Exact substitutability is rare – e.g. cat/dog

Limitations

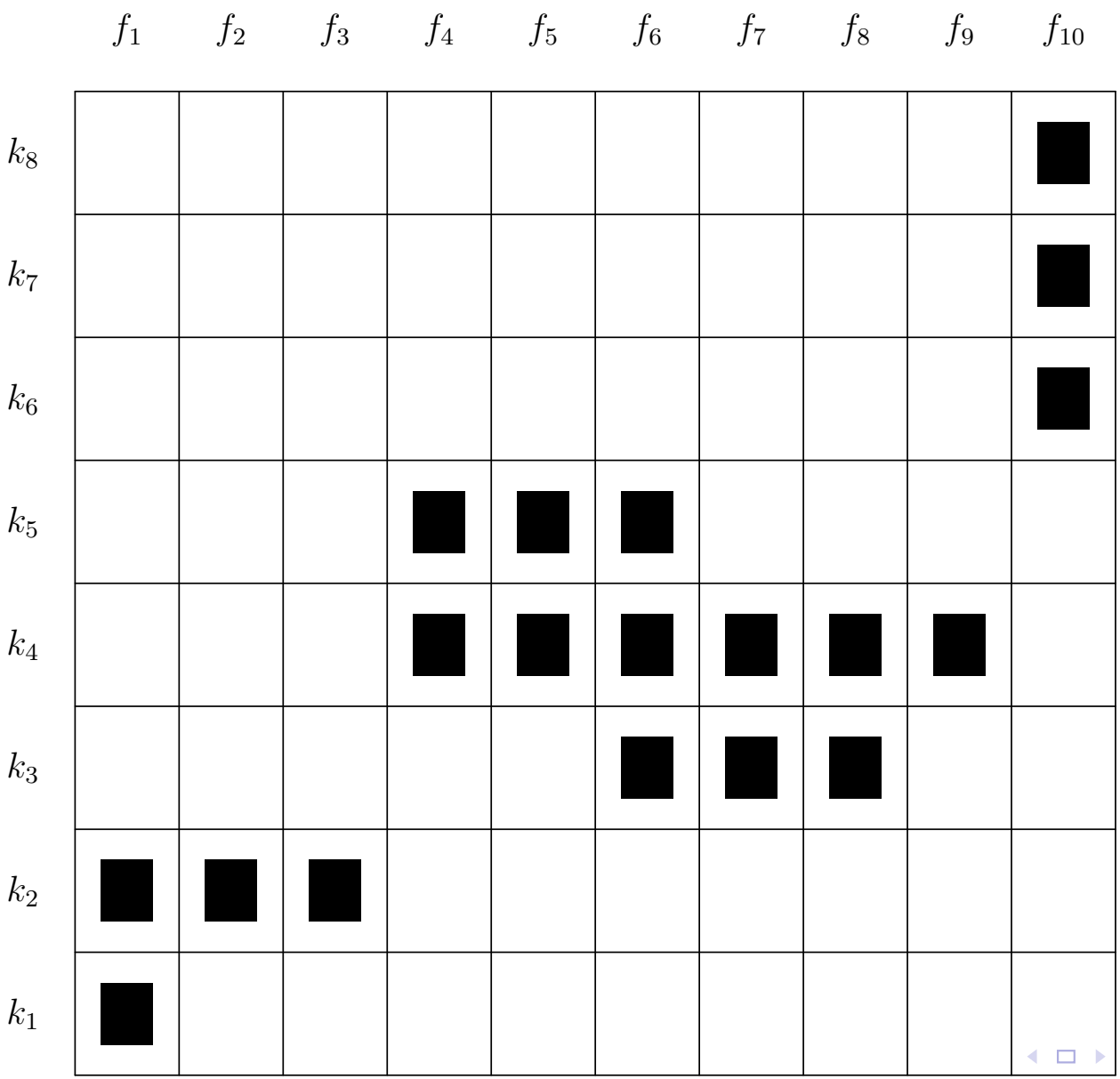
One symbol per congruence class just won't work for natural languages:

- Congruence classes are very many and very close together
- Exact substitutability is rare – e.g. cat/dog
- Learning model assumes that either they are identical or they are completely unrelated.
- Need to have a more powerful representation that represents the structure of the congruence classes
- Languages aren't context free

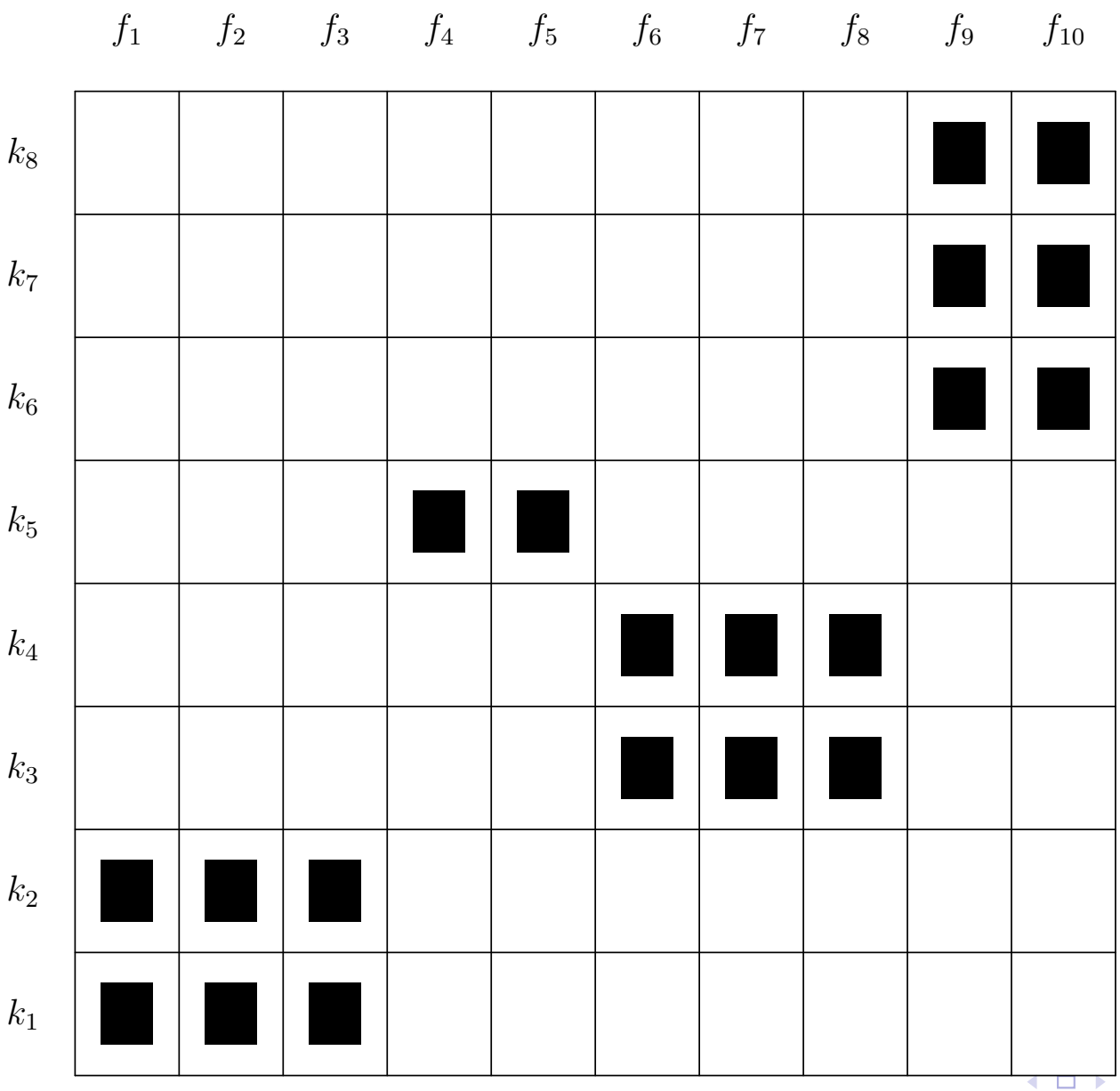
Observation table
filled in with MQs

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
k_8										
k_7										
k_6										
k_5										
k_4										
k_3										
k_2										
k_1										

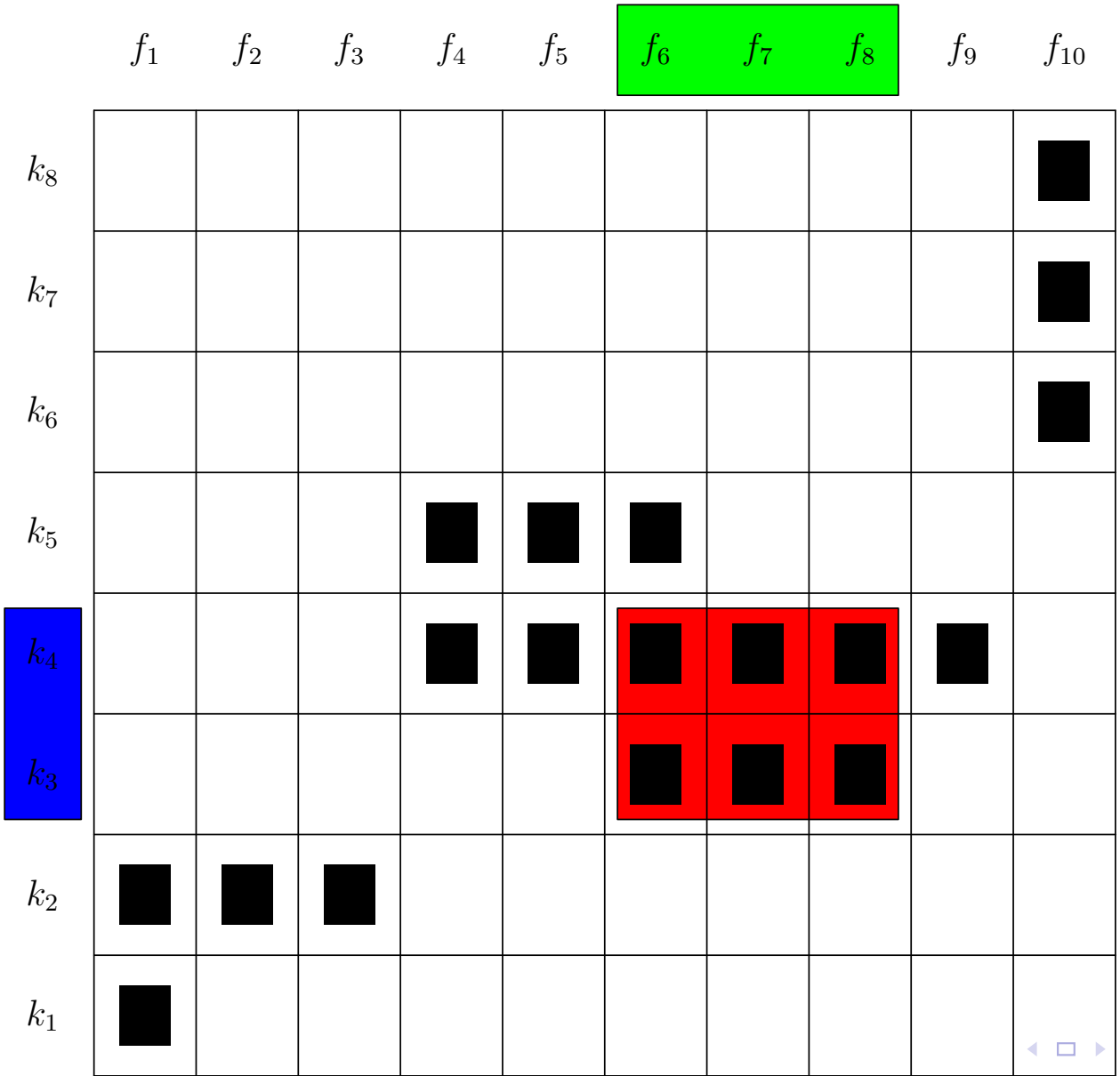
Observation table
filled in with MQs



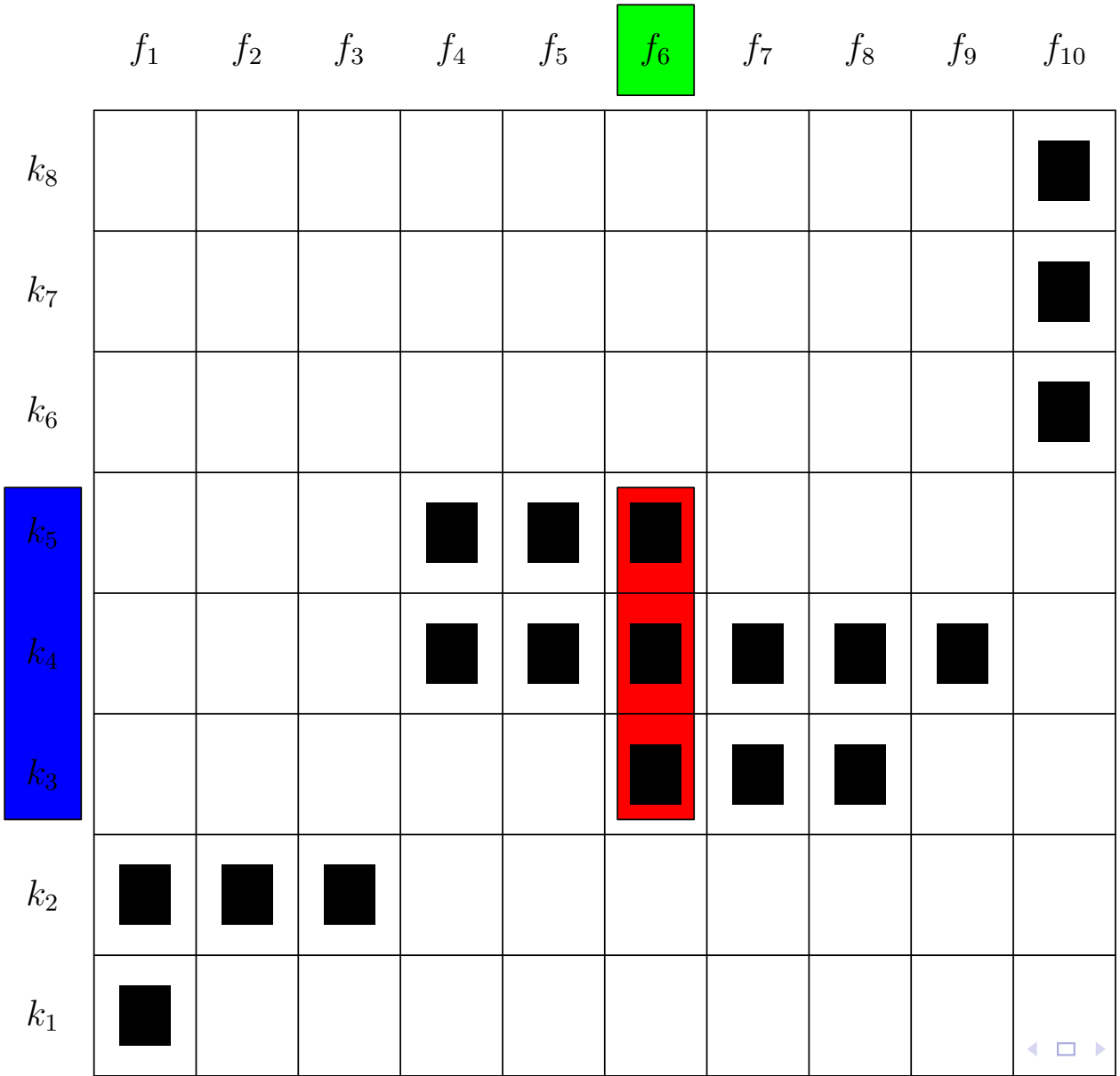
Substitutable



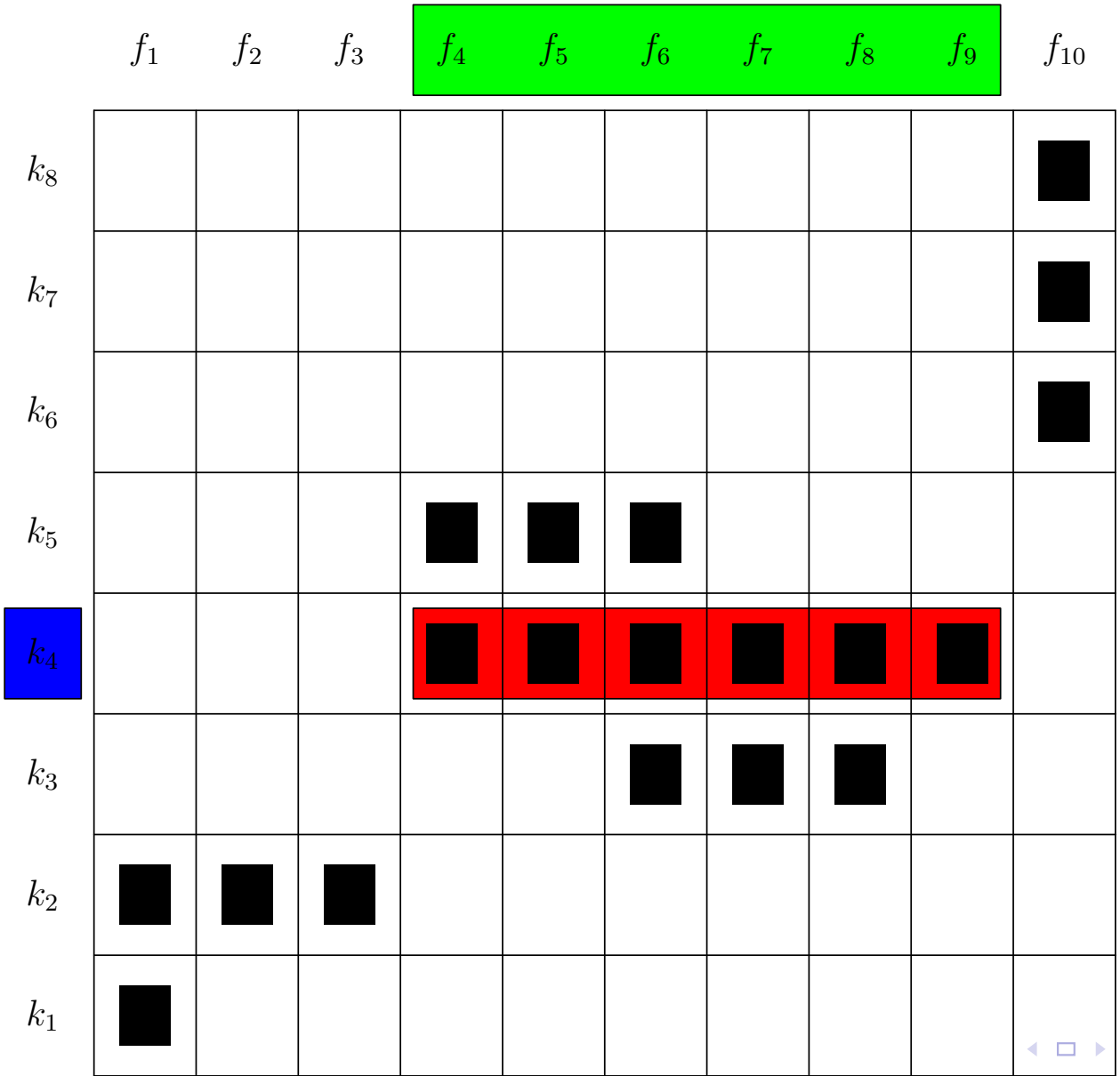
Concepts
Maximal rectangles



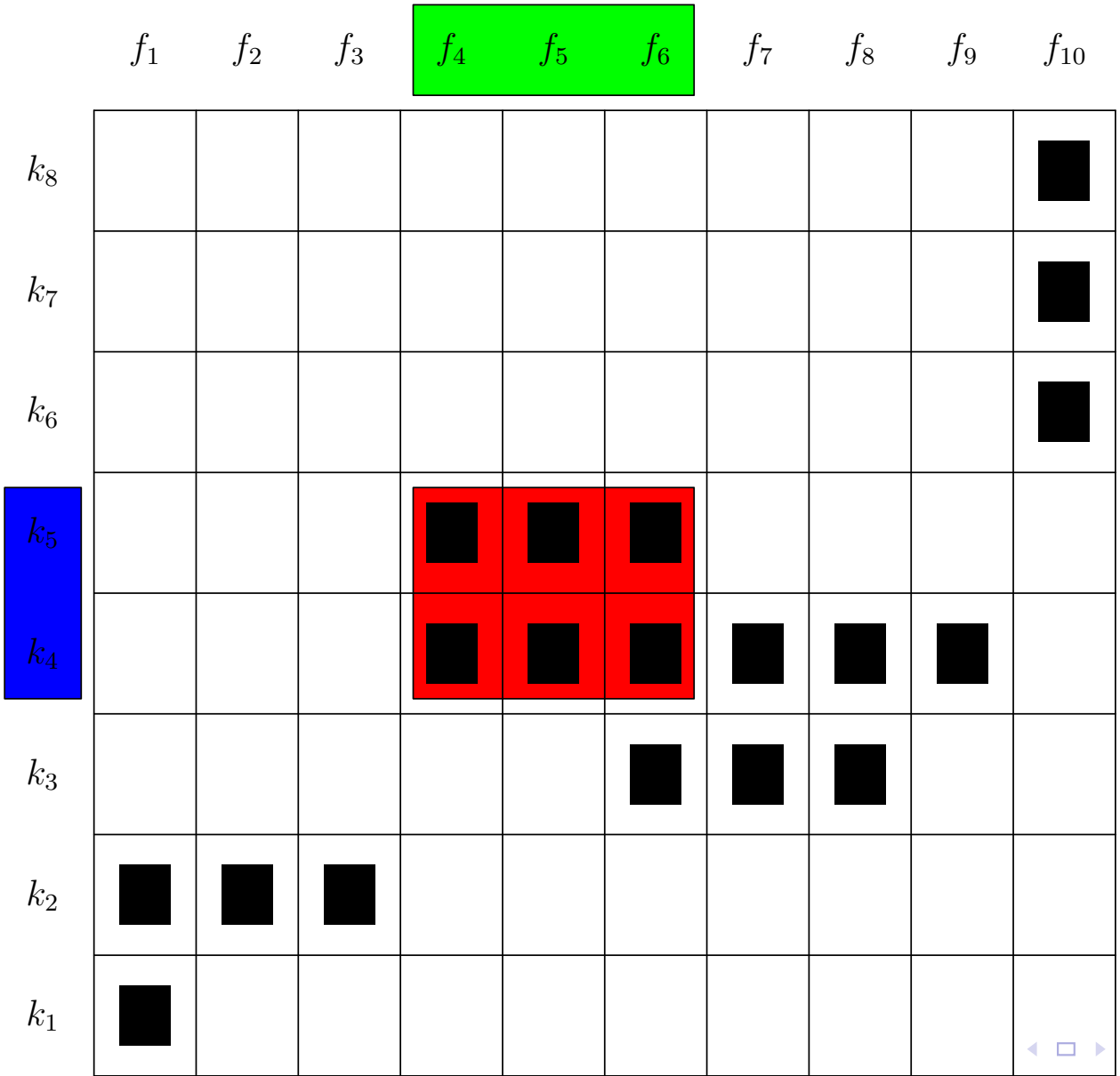
Concepts
Maximal rectangles



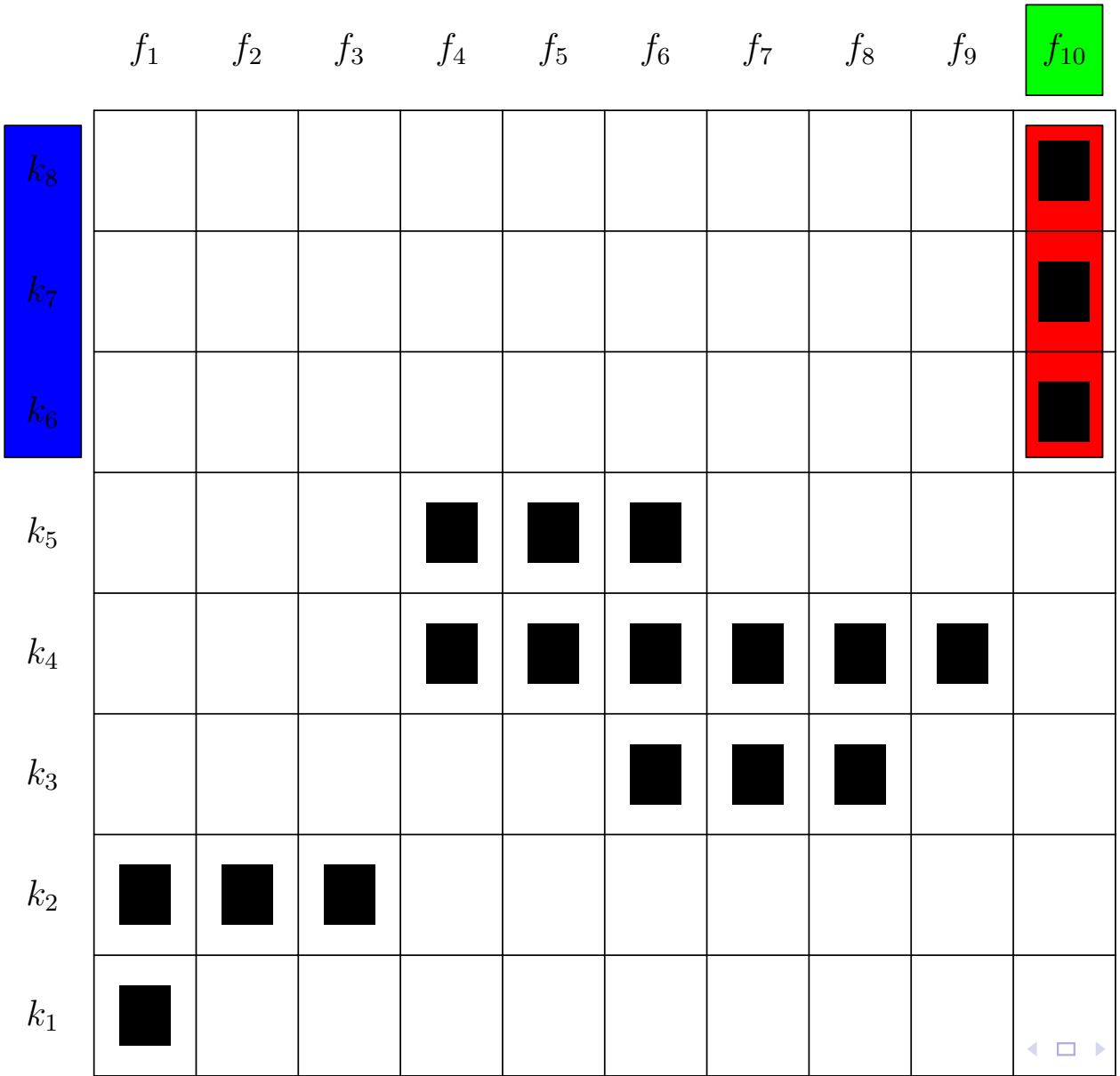
Concepts
Maximal rectangles



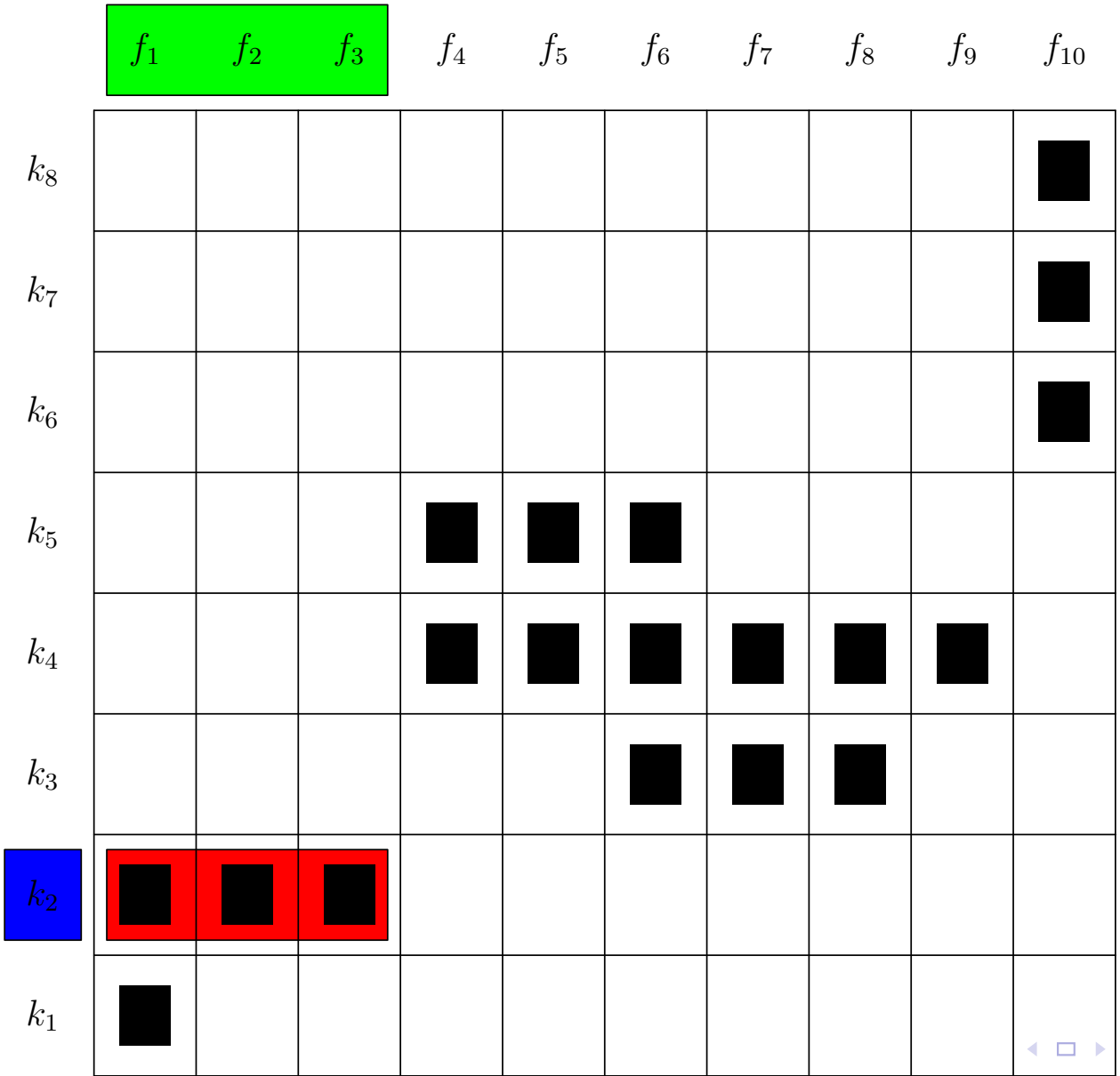
Concepts
Maximal rectangles



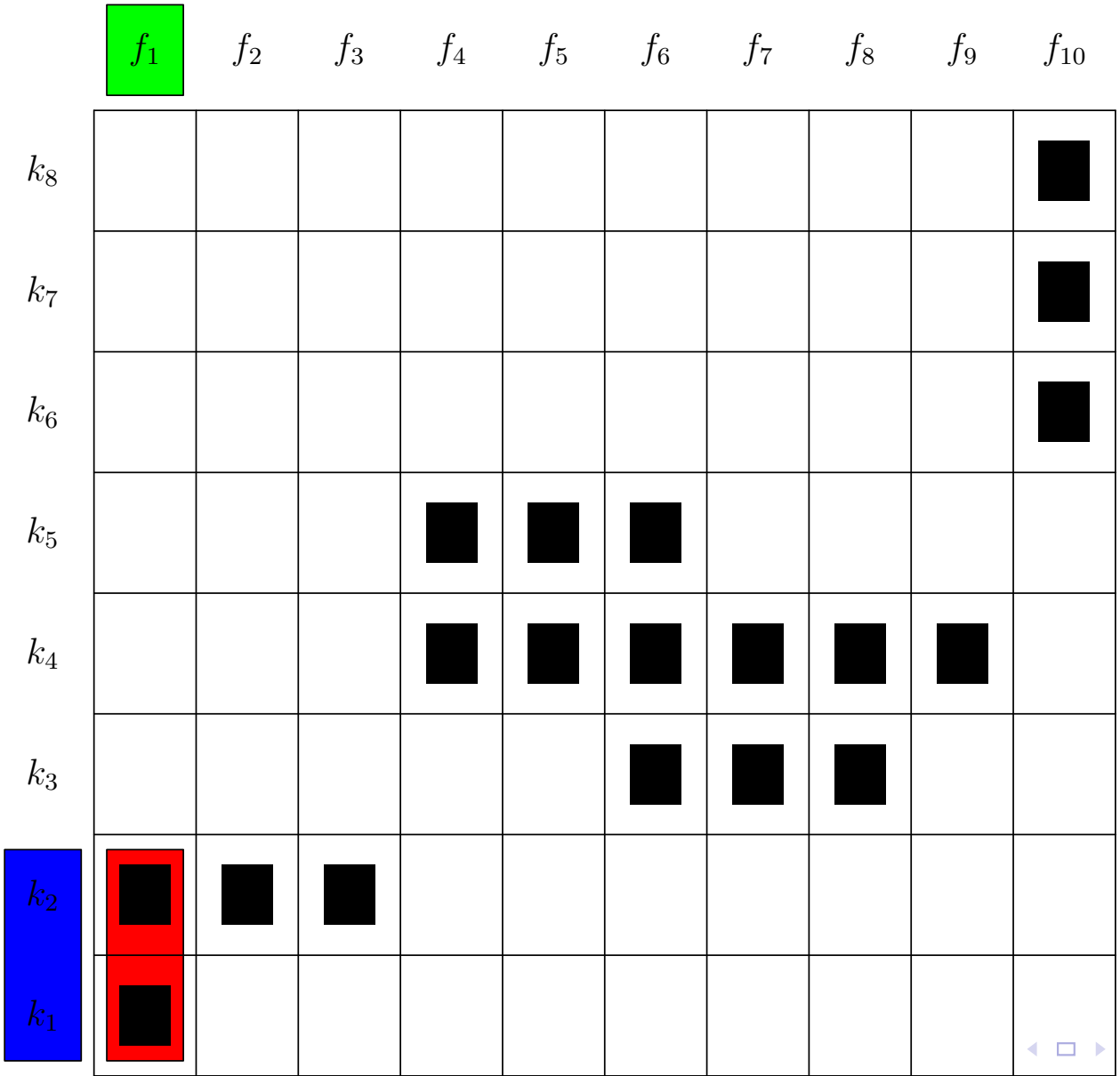
Concepts
Maximal rectangles



Concepts
Maximal rectangles



Concepts
Maximal rectangles



Relation to CFGs

Define

Given a CFG G for each non-terminal N

- Yield: $Y(N) = \{w | N \stackrel{*}{\Rightarrow} w\}$
- Context: $C(N) = \{(l, r) | S \stackrel{*}{\Rightarrow} lNr\}$.

Clearly $C(N) \odot Y(N) \subseteq L$
Each non-terminal will be a rectangle – but not necessarily maximal.

Formally

Polar maps

$S' = \{(l, r) \in F : \forall w \in S \;\; lwr \in L\}$
$C' = \{w \in K : \forall (l, r) \in C \;\; lwr \in L\}$

Concept

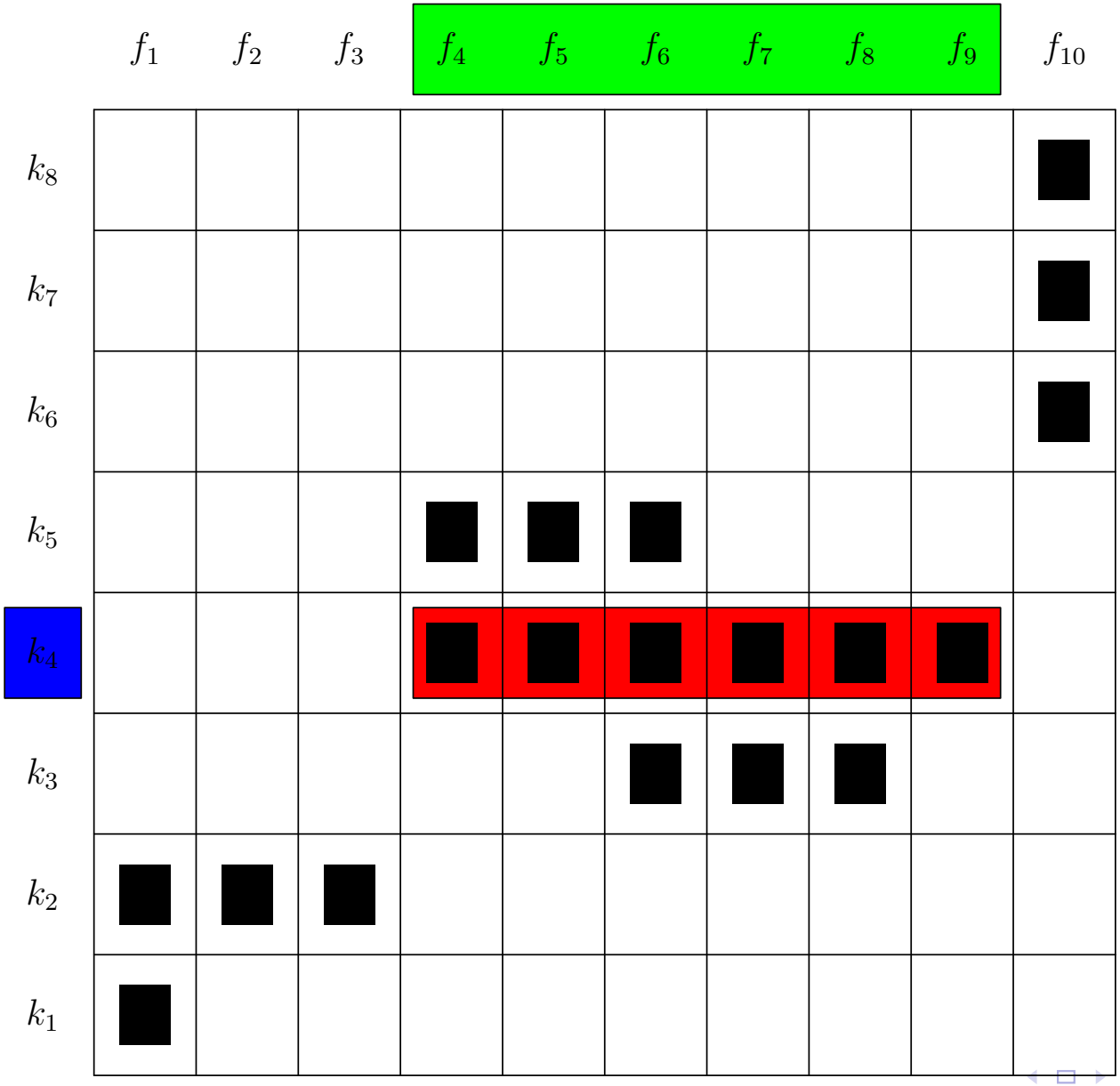
Ordered pair $\langle S, C \rangle$

- $S \subseteq K$ the set of strings
- $C \subseteq F$ is a set of contexts

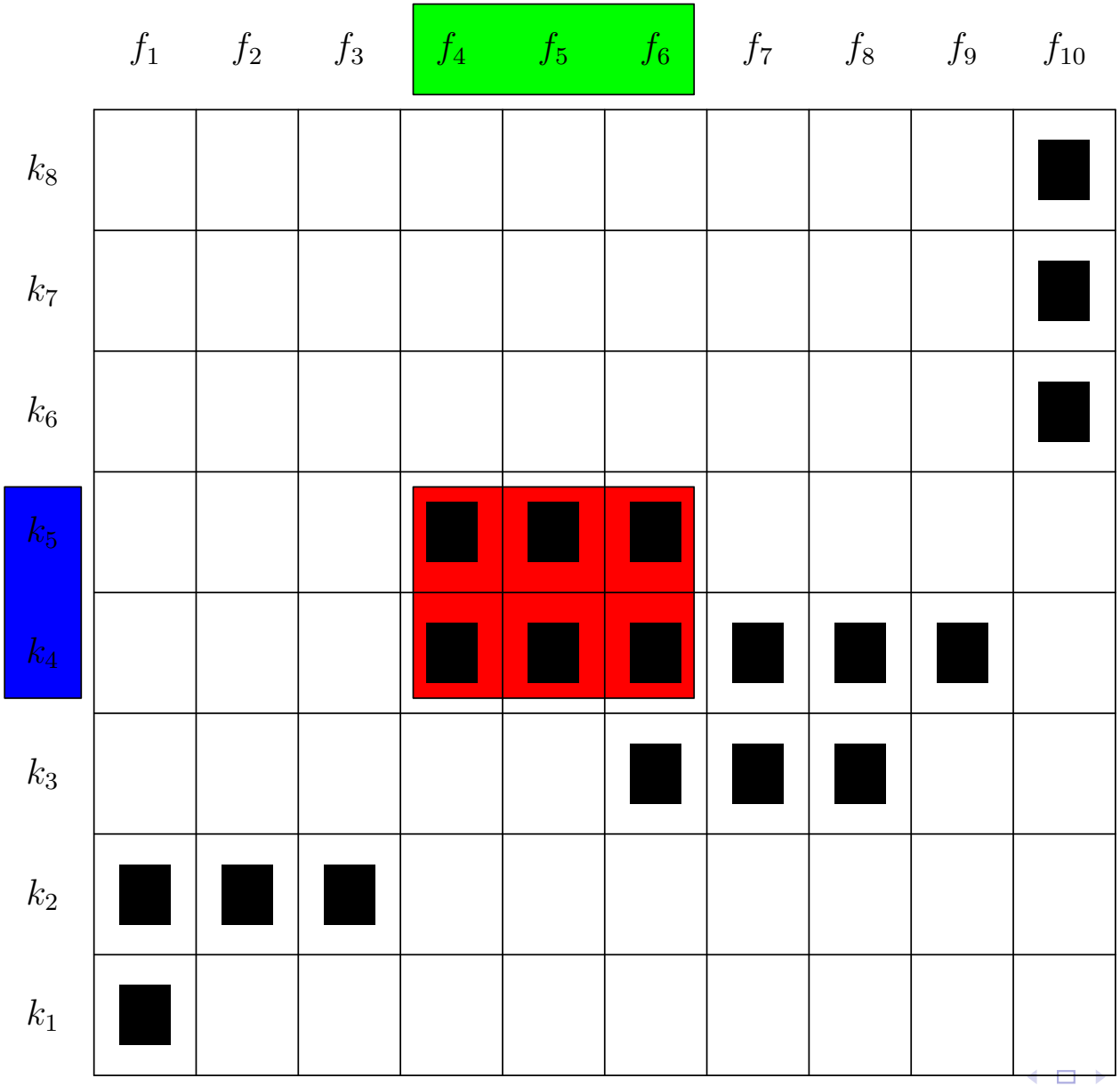
$S' = C$ and $C' = S$

$\mathcal{C}(S) = \langle S'', S' \rangle$

Partial order

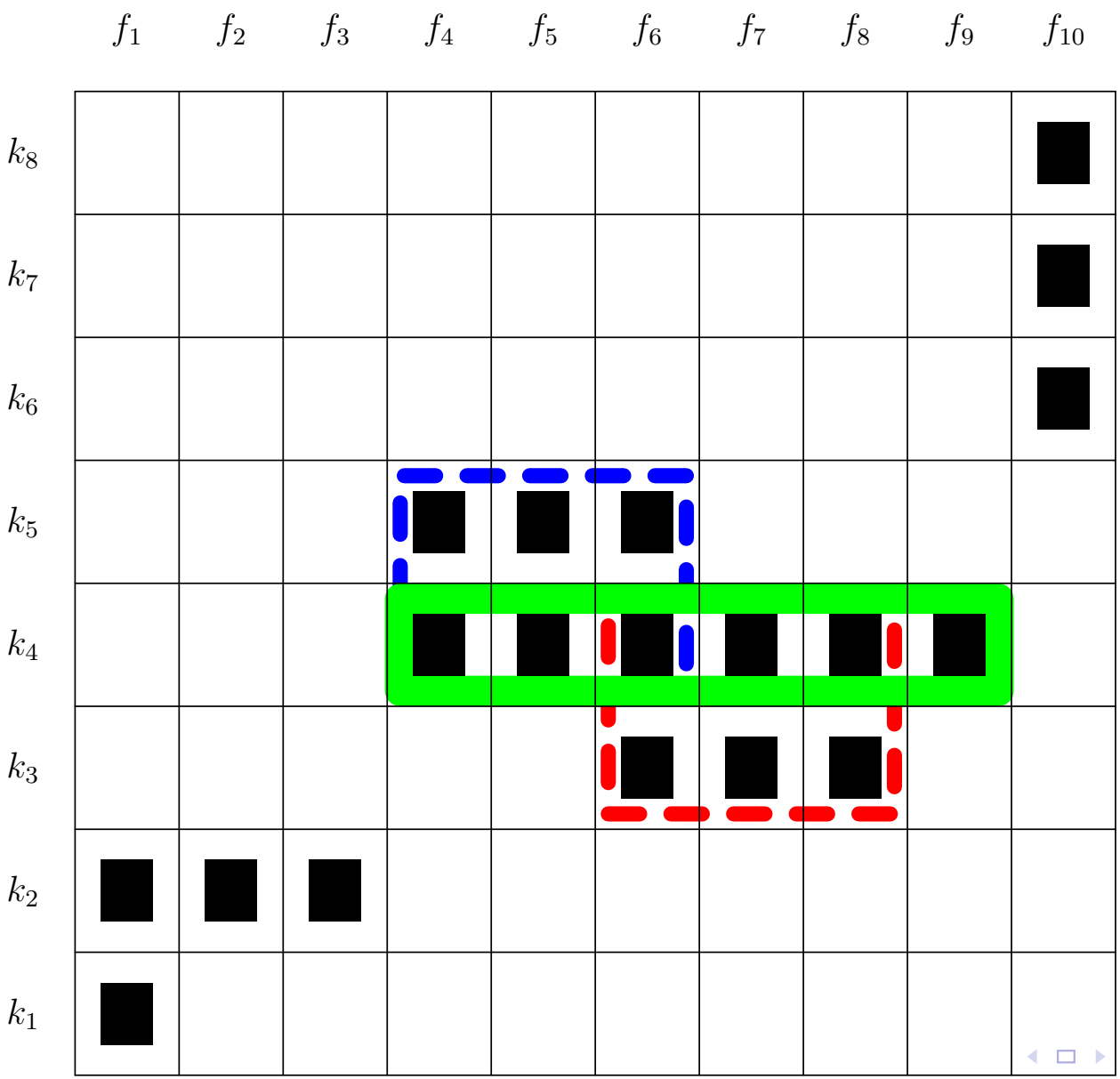


Partial order



Greatest lower bound

Meet or $X \wedge Y$



Formally

Partial order

$\langle S_x, C_X \rangle \leq \langle S_Y, C_Y \rangle$
iff $S_X \subseteq S_Y$ (or $C_X \supseteq C_Y$)

Meet

$\langle S_x, C_X \rangle \wedge \langle S_Y, C_Y \rangle$
is $\langle S_x \cap S_y, (S_x \cap S_y)' \rangle$

Concept lattice: Formal Concept Analysis

The set of all concepts form a complete lattice $\mathfrak{B}(K, D, F)$
Top $\top = \langle K, \emptyset \rangle$
Bottom $\bot = \langle \emptyset, F \rangle$

Concatenation

Two concepts

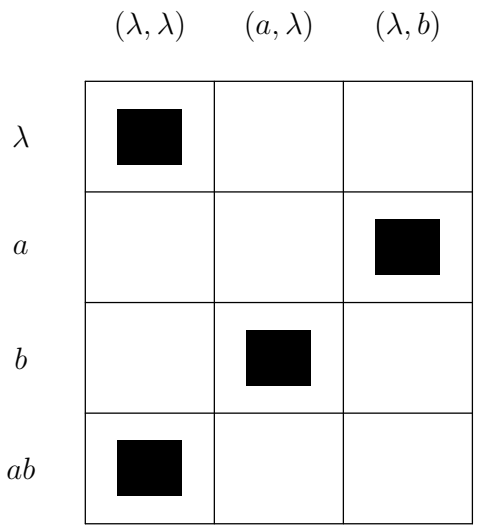
$\langle S_x, C_x \rangle \circ \langle S_Y, C_Y \rangle$
Concatenate the sets of strings $S_x S_y$: a subset of KK
Take the set of contexts $(S_X S_Y)'$ and make concept from that.
$\langle (S_x S_Y)'', (S_X S_Y)''' \rangle$

Data

Increase D to be a subset of $F \odot KK$
$\{luvr | (l, r) \in F, u, v \in K\}$

Dyck language

λ, ab, $abab$, $aabb$, $abaabb$ . . .



- $L = \langle \{\lambda, ab\}, (\lambda, \lambda) \rangle$
- $A = \langle \{a\}, (\lambda, b) \rangle$
- $B = \langle \{b\}, (a, \lambda) \rangle$
- $\top$
- $\bot$

Dyck language

$L \circ L$

$S_X = S_Y = \{\lambda, ab\}$
$S_X S_Y = \{\lambda, ab, abab\}$
$(S_X S_Y)' = (\lambda, \lambda)$
Result is L

$B \circ A$

$S_X = \{b\}, S_Y = \{a\}$
$S_X S_Y = \{ba\}$
$(S_X S_Y)' = \emptyset$
Result is $\top$

Dyck language

λ, *ab*, *abab*, *aabb*, *abaabb* . . .

- $L = \langle \{\lambda, ab\}, (\lambda, \lambda) \rangle$
- $A = \langle \{a\}, (\lambda, b) \rangle$
- $B = \langle \{b\}, (a, \lambda) \rangle$
- $\top$
- $\bot$

	$\top$	L	A	B	$\bot$
$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$
L	$\top$	L	A	B	$\bot$
A	$\top$	A	$\top$	L	$\bot$
B	$\top$	B	$\top$	$\top$	$\bot$
$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$

Constructing a representation

Natural question

We have this concatenation operation:
How do we use this to infer a representation?
A CFG, a TAG etc

Constructing a representation

Natural question

We have this concatenation operation:
How do we use this to infer a representation?
A CFG, a TAG etc

This lattice is a representation already!

Recursive definition

$u = a_1 \dots a_k$

- If we know $\mathcal{C}(a)$ for all $a \in \Sigma$ and we can concatenate the concepts
- Then we can compute $\mathcal{C}(a_1 a_2 \dots a_k)$ as $\mathcal{C}(a_1) \circ \mathcal{C}(a_2) \dots \mathcal{C}(a_k)$
- If $\mathcal{C}(u)$ contains (λ, λ) then it is in the language.

Representation

Compute approximation to distribution

$\phi_G : \Sigma^* \rightarrow \mathfrak{B}(K, D, F)$

- $\phi(a) = \mathcal{C}(a)$ — look it up
- $\phi(ab) = \phi(a) \circ \phi(b)$
- etc

A problem

Not associative

aab

- $\mathcal{C}(a) \circ (\mathcal{C}(a) \circ \mathcal{C}(b)) = \mathcal{C}(a)$
- $(\mathcal{C}(a) \circ \mathcal{C}(a)) \circ \mathcal{C}(b) = \top \circ \mathcal{C}(b) = \top$

Combining

We want the best possible estimate – the one with the most contexts:

- If one bracketing gives X and another gives Y
- we can predict $X \wedge Y$

Representation

Definition

A distributional lattice grammar (DLG) is a tuple $\langle K, D, F \rangle$ where

- K is a finite subset of strings that includes Σ and λ
- F is a finite set of contexts that includes (λ, λ)
- D is a finite subset of $F \odot KK$

Example

DLG for the Dyck language:

- $K = \{\lambda, a, b, ab\}$
- $F = \{(\lambda, \lambda), (a, \lambda), (\lambda, b)\}$
- $D = \{\lambda, ab, abab, aabb\}$

Representation

Definition

$\phi : \Sigma^* \to \mathfrak{B}(K, D, F)$.

- $\phi(\lambda) = \mathcal{C}(\lambda)$
- for all $a \in \Sigma$, (i.e. for all $w, |w| = 1$)
  $\phi(a) = \mathcal{C}(a)$
- for all w with $|w| > 1$,

$$\phi(w) = \bigwedge_{u,v \in \Sigma^+ : uv = w} \phi(u) \circ \phi(v)$$

Language

Define $L(\mathfrak{B}(K, L, F)) = \{w : \phi(w) \leq \mathcal{C}(\{(\lambda, \lambda)\})\}$

Dyck language

- $\phi(a) = A, \phi(b) = B$
- $\phi(aa) = \top, \phi(ab) = L, \ldots$
- $\phi(aab) = \phi(aa) \circ \phi(b) \wedge \phi(a) \circ \phi(ab) = A$
- $\phi(abab) = \ldots$
- $\phi(abababab) = \phi(a) \circ \phi(bababab) \wedge \cdots = L$

$\phi(w)$ has feature (λ, λ) iff w is in Dyck language.

Learnability

Data

Given a set K and some context F, we can figure out which elements of $F \odot KK$ are in L.
Probabilistically or not . . .

Search

How can we find suitable K and F?

Learnability

Data

Given a set K and some context F, we can figure out which elements of $F \odot KK$ are in L.
Probabilistically or not ...

Search

How can we find suitable K and F?

Notation

$D = L \cap F \odot KK$
$\langle K, L \cap F \odot KK, F \rangle = \langle K, L, F \rangle$

Change the set of strings

$J \subseteq K$

Map

g from $\mathfrak{B}(J, L, F)$ to $\mathfrak{B}(K, L, F)$ (from the smaller lattice to the larger lattice) as $g(\langle S, C \rangle) = \langle C', C \rangle$.

Change the set of strings

$J \subseteq K$

Map

g from $\mathfrak{B}(J, L, F)$ to $\mathfrak{B}(K, L, F)$ (from the smaller lattice to the larger lattice) as $g(\langle S, C \rangle) = \langle C', C \rangle$.

Lemma

$g(\phi_J(w)) \leq \phi_K(w)$

Increasing K

- $g(\phi_J(w)) \leq \phi_K(w)$ means that as we increase K the language defined by $\langle K, L, F \rangle$ decreases monotonically
- After a finite number of strings it will converge
- It will always converge to a subset of L

(Intuitively, as the strings increase the sets of contexts shared decrease)

Power of Representation

Language class

Let $\mathcal{L}$ be the set of all languages L such that there is a *finite* set of contexts F s.t. $L = L(\mathfrak{B}(\Sigma^*, L, F))$

Includes

1. All regular languages
2. Some but not all CFLs
3. Some non context free languages

Change the set of contexts

$F \subseteq G$

Map

f from $\mathfrak{B}(K, L, G)$ to $\mathfrak{B}(K, L, F)$, (from larger to smaller) as $f(\langle S, C \rangle) = \langle (C \cap F)', C \cap F \rangle$.

Change the set of contexts

$F \subseteq G$

Map

f from $\mathfrak{B}(K, L, G)$ to $\mathfrak{B}(K, L, F)$, (from larger to smaller) as $f(\langle S, C \rangle) = \langle (C \cap F)', C \cap F \rangle$.

Lemma

$f(\phi_G(w)) \leq \phi_F(w)$

As we increase the set of contexts the language monotonically increases.

Search problem is trivial

Naive Algorithm

Start with $F = \{(\lambda, \lambda)\}, K = \Sigma \cup \{\lambda\}$

- If we see a string that is not in our hypothesis, the hypothesis is too small, and we add contexts to F
- Add strings to K if it will change the lattice at all.

Clark, (CoNLL, 2010)

DLGs can be learnt from positive data and MQs
Polynomial update time

Context sensitive example

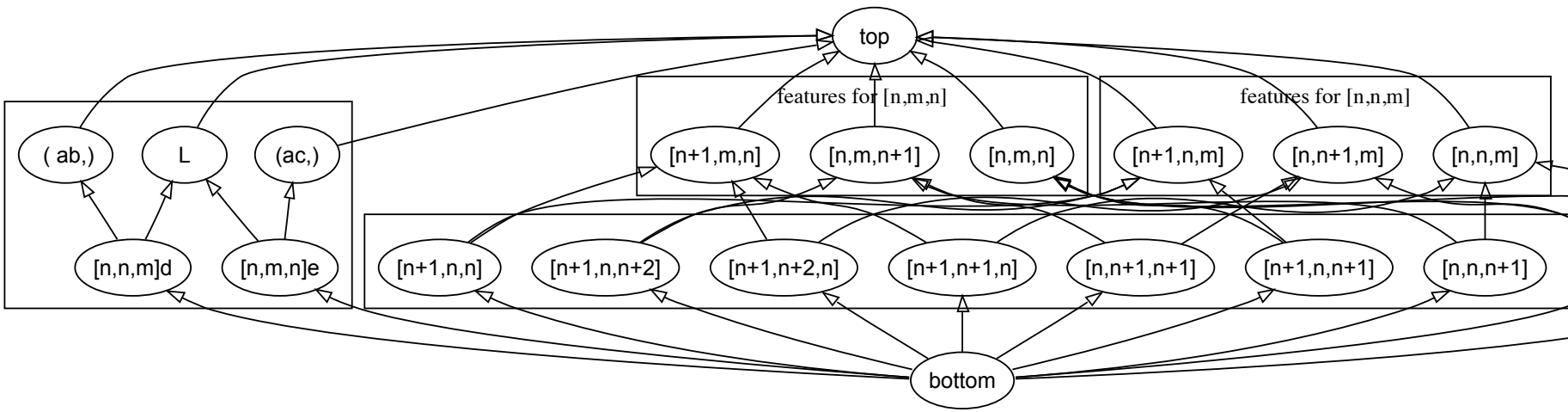
MIX language (Bach, 1981)

$\{w \in \{a, b, c\}^* || w|_a = |w|_b = |w|_c\}$

DLG example

Let $M = \{(a, b, c)^*\}$, we consider the language
$L = L_{abc} \cup L_{ab} \cup L_{ac}$ where $L_{ab} = \{wd | w \in M, |w|_a = |w|_b\}$,
$L_{ac} = \{we | w \in M, |w|_a = |w|_c\}$,
$L_{abc} = \{wf | w \in M, |w|_a = |w|_b = |w|_c\}$.

Lattice


top
features for [n,m,n]
features for [n,n,m]
(ab,)
L
(ac,)
[n+1,m,n]
[n,m,n+1]
[n,m,n]
[n+1,n,m]
[n,n+1,m]
[n,n,m]
[n,n,m]d
[n,m,n]e
[n+1,n,n]
[n+1,n,n+2]
[n+1,n+2,n]
[n+1,n+1,n]
[n,n+1,n+1]
[n+1,n,n+1]
[n,n,n+1]
bottom

Context free languages

Finite Context Property

For a non-terminal N, if there is a finite set of contexts F_N such that $L(G, N) = F'_N$, then it has the FCP.
CFGs with the FCP are representable by DLGs.

Not in DLG?

$L = \{a^n b | n > 0\} \cup \{a^n c^m | m > n > 0\}$

Switch to non context-free representation

CFG inference idea

Build a CFG with one non-terminal for each concept
$C[A] = \bigcap_{(l,r)\in A} C[l, r]$

Problem: exponentially many non-terminals – we can't parse.

Parsing

We can lazily represent the huge CFG but we cannot parse
Note: if $C[A] \stackrel{*}{\Rightarrow} w$ and $C[B] \stackrel{*}{\Rightarrow} w$
then $w \in C[A] \cap I[B] = C[A \cup B]$
Take the union (roughly) of all the sets and combine them.

Shift to a CS representation for efficient computation!

Outline

A serious criticism

Montague

"Syntax is only interesting as a precursor to semantics"

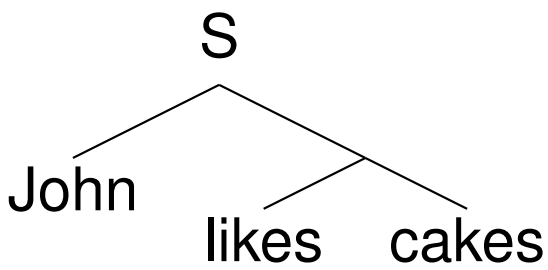
- These models just tell you whether a string is in the language or not, which is useless and "irrelevant".
- The standard assumption is that each sentence has a hidden tree structure that we need to recover to do semantics.
- We need to learn this constituent structure

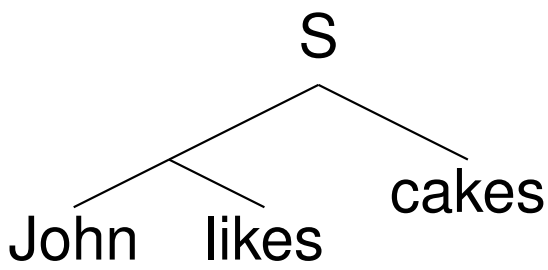
Associativity

Formally the key point about hierarchical representations is that they are not *associative*.

- $x + (y + z) = (x + y) + z$
- String concatenation is associative – *abc*
- Tree operations are not: $(a(bc)) \neq ((ab)c)$

Obligatory choice





Free word order in Finnish

"Anna gets flowers"

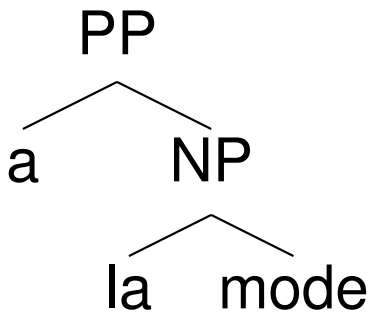
- Anna sai kukkia
- Anna kukkia sai
- sai kukkia Anna
- sai Anna kukkia
- kukkia sai Anna
- kukkia Anna sai

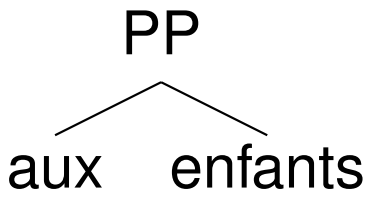
Bracketing paradoxes

Sproat, 1992

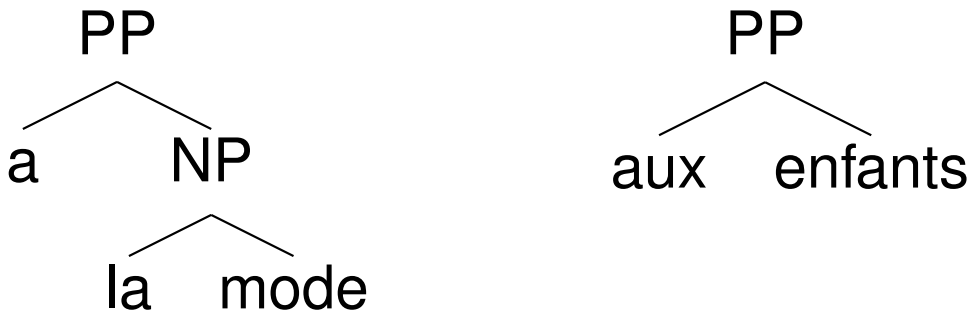
- unrulier
- [*un*[*rulier*]] or [[*unruli*]*er*]

Romance clitics



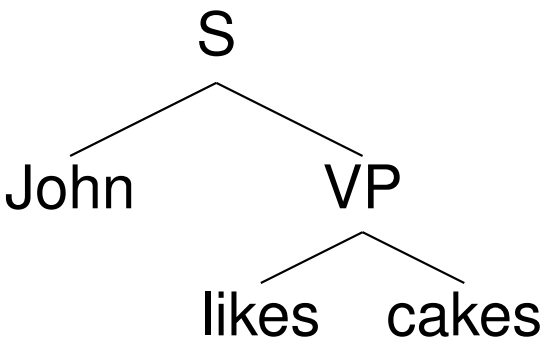


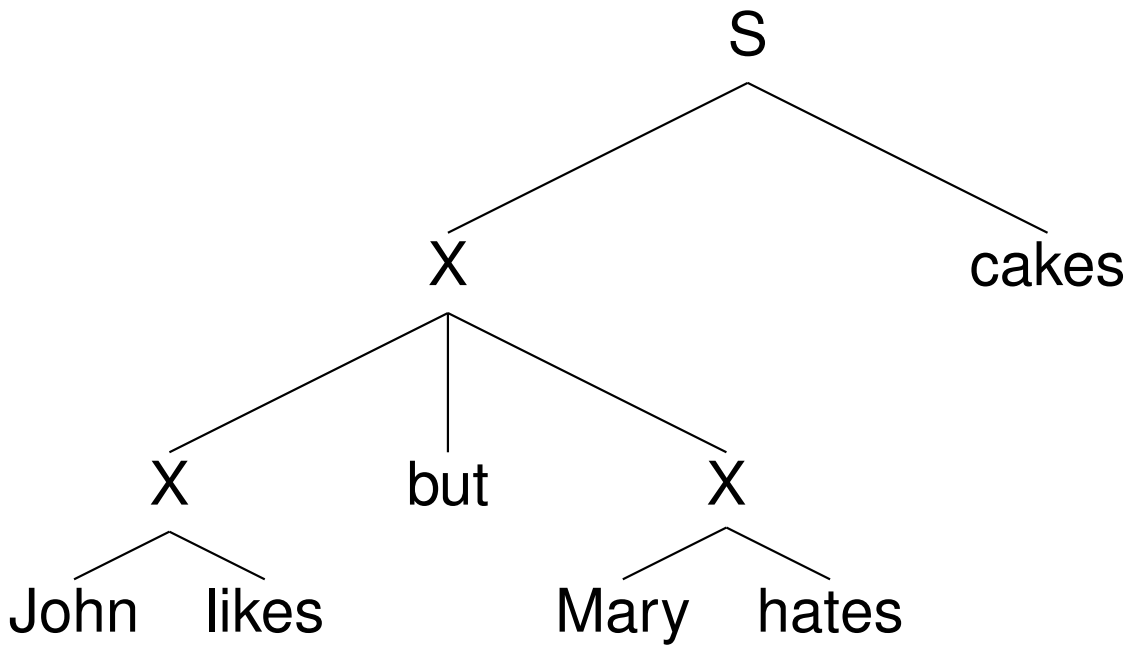
Romance clitics



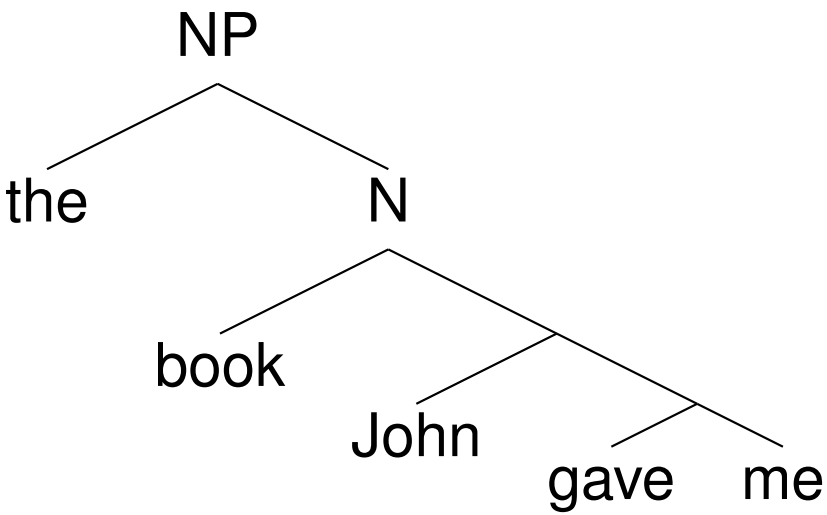
- Italian: glielo, andarsene
- German: an dem Tisch → am Tisch

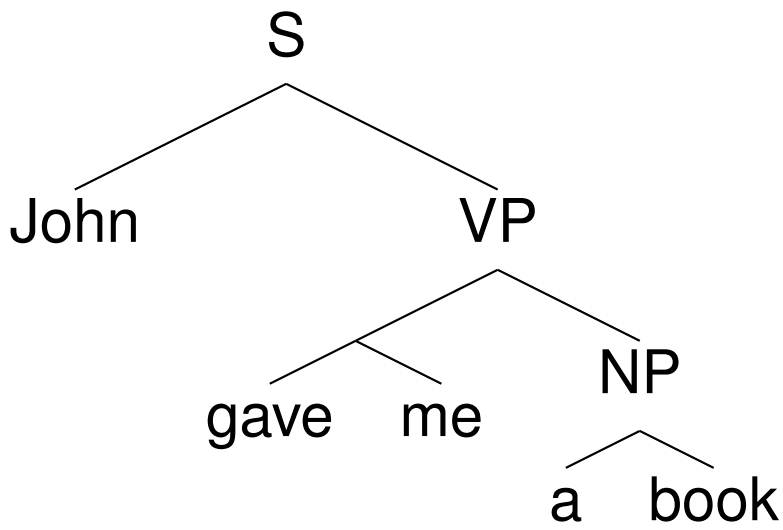
Non constituent coordination





Movement

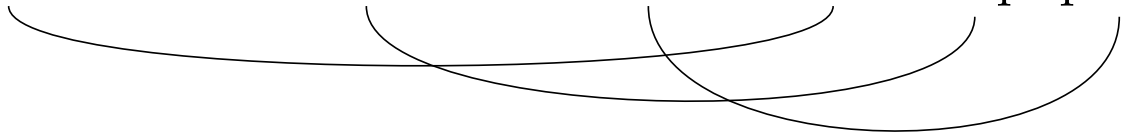




Cross-serial dependencies

Shieber 82

... das mer d'chind em Hans es huus lönd hälfe aastriiche
... that we the children-ACC Hans-DAT house-ACC let help paint



'... that we let the children help Hans paint the house'

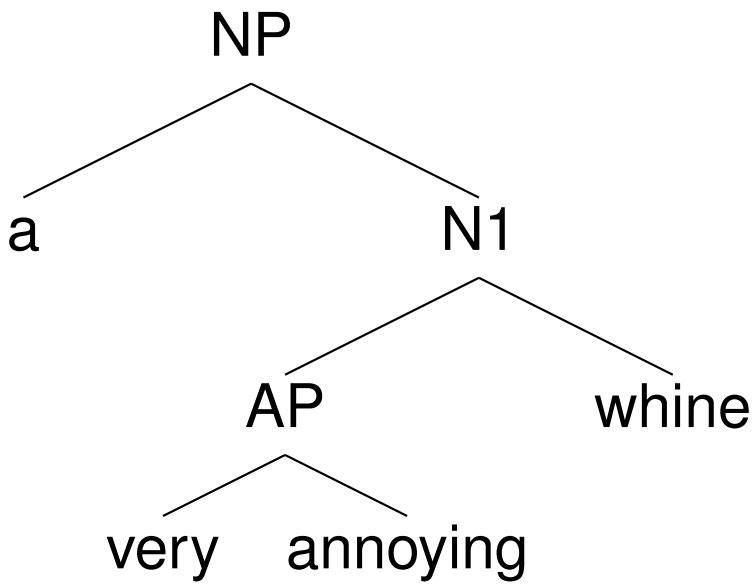
Phonology

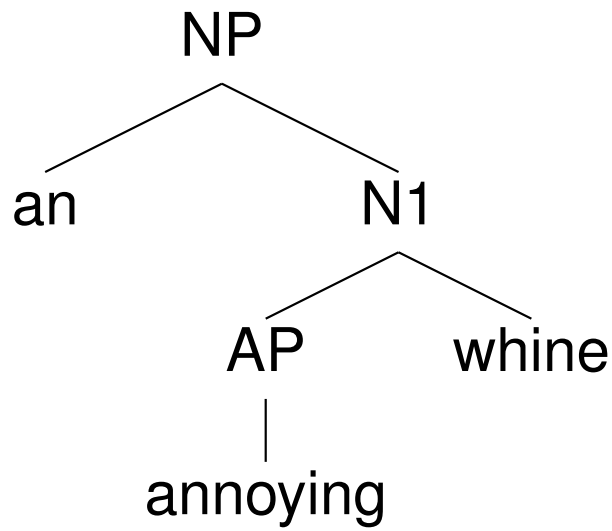
Syllables don't coincide with words

Liaison and enchainement in French

pətɪtãfã
petit enfant
(small child)

Alternations





- Italian: il/lo; Slovenian: s/z . . .

Etc.

- Tmesis
- Interjections
- Discontinuous constituents
- German scrambling
- . . .

Standard linguistic counter-arguments

None of these arguments are original or fatal;

- Rigid tree structure is not overt
- Allow movement of words and phrases from place to place
- Allow a rich variety of phonologically null constituents
- Distinction between levels of phonology and syntax each with separate trees
- Use TAG based formalisms with a richer set of tree operations
- Core-periphery distinction

Uncontroversial

[Bouma, 1989] These and other such arguments suggest that there is no such thing as a fixed constituent structure, but that the order in which elements combine with each other is rather free.

Uncontroversial

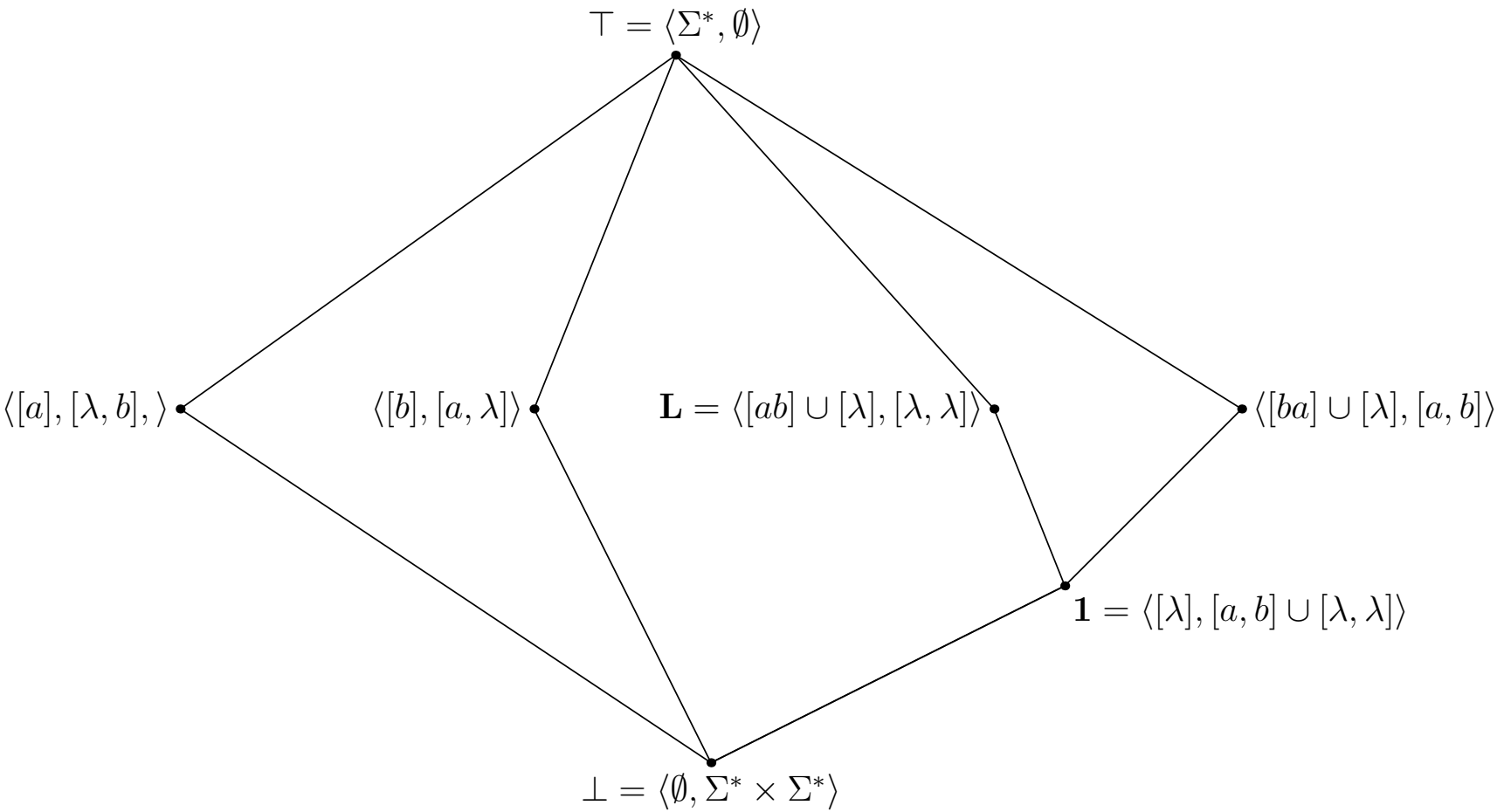
[Bouma, 1989] These and other such arguments suggest that there is no such thing as a fixed constituent structure, but that the order in which elements combine with each other is rather free.

- Many theories allow flexible constituency: multiple trees for the same unambiguous sentence
- Structural completeness is an advantage for processing.
- CCG (Steedman, 1997)
- Dependency grammar
- Associative Lambek calculus

The Full lattice

- Suppose K is every string
- Suppose F is every context $\Sigma^* \times \Sigma^*$
- We have the lattice $\mathfrak{B}(L)$

$L = (ab)^*$



Residuated lattice

This is a complete residuated lattice; written $\mathfrak{B}(L)$.
Basis of CG which has the cleanest syntax-semantics interface.

- Concatenation is a monoid: associative and with unit $\mathcal{C}(\{\lambda\})$.
- Lattice : $X \wedge Y$ is a greatest lower bound, $X \vee Y$ is a least upper bound; $X \leq Y$ is a partial order.
- The two operations interact properly, and we have binary operations $/$, and $\backslash$ such that
  $X \circ Y \leq Z$ iff $X \leq Z/Y$ iff $Y \leq X\backslash Z$

Idea for structural descriptions

"Parse trees"

We have a recognizer but we want a parser:

Admissible structures for a string w

Each span has a concept $\psi[i, j]$

- $\psi[i, j] \geq \mathcal{C}(w[i : j])$
- $\psi[i, j] \geq \bigwedge_k \psi[i, k] \circ \psi[k, j]$
- $\psi[0, l] \leq \mathcal{C}(L)$

Maximal structures

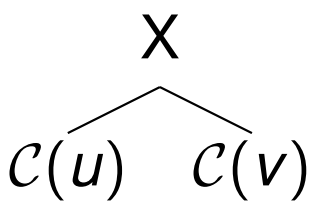
The set of maximal structures under the natural partial order can be viewed as the set of structural descriptions.
Discard $\top$ symbols and construct a graph or DAG.

Idea for structural descriptions

Spurious ambiguity

Structural completeness

If we have the full lattice, then we can have any binary tree.

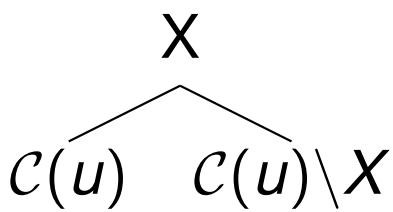


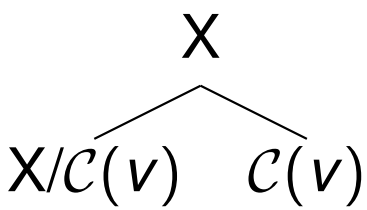
Idea for structural descriptions

Spurious ambiguity

Structural completeness

If we have the full lattice, then we can have any binary tree.



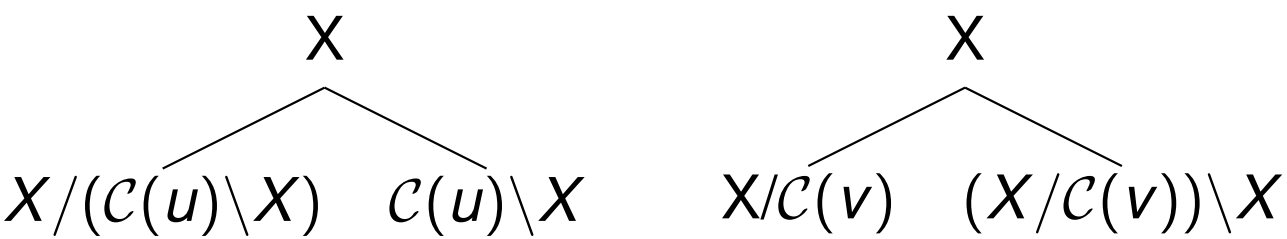


Idea for structural descriptions

Spurious ambiguity

Structural completeness

If we have the full lattice, then we can have any binary tree.



Why the delay?

The perceived problem

Not enough to learn a grammar, you have to learn the right grammar.
You have to learn constituent structure
Distributional structure is a product of hidden constituent structure.

Why the delay?

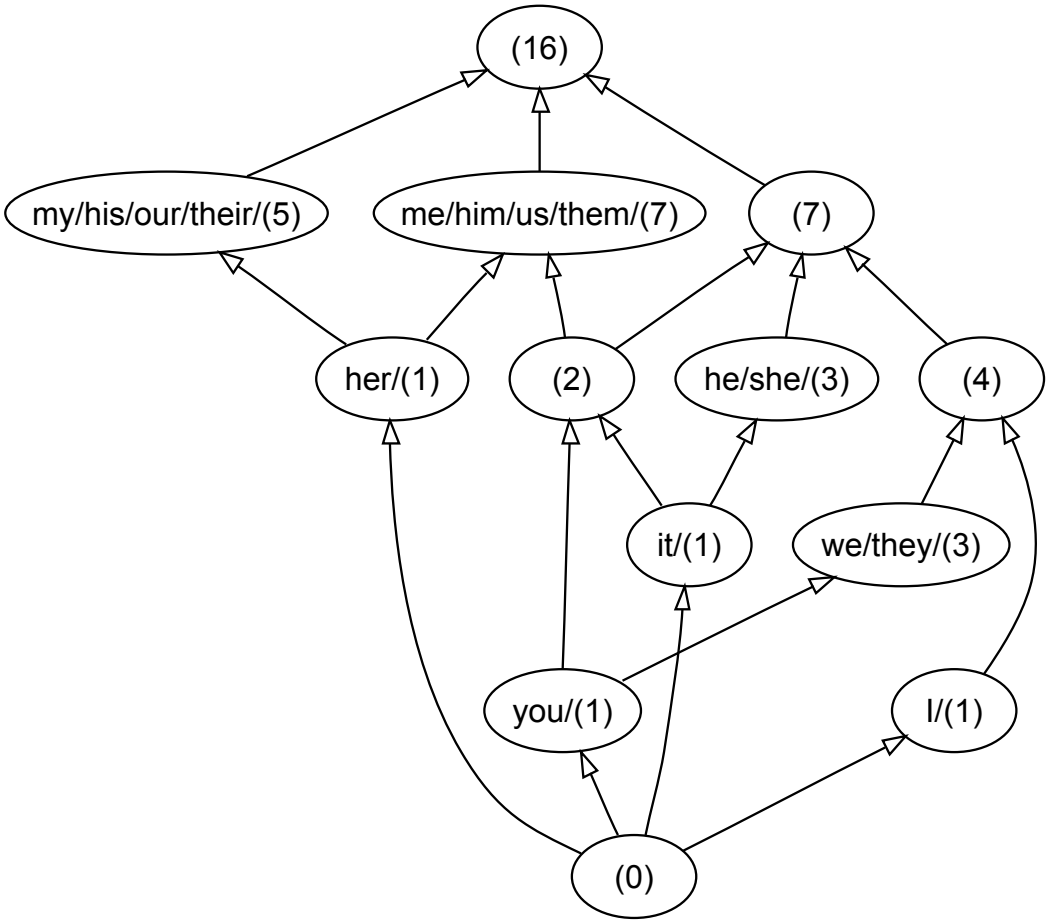
The perceived problem

Not enough to learn a grammar, you have to learn the right grammar.
You have to learn constituent structure
Distributional structure is a product of hidden constituent structure.

Maybe distributional structure is all there is?
If we can

- represent (weak) syntax
- learn
- support semantic interpretation?

Linguistic concepts



Outline

Conclusion

Jackendoff (2008)

1. Descriptive constraint: the class of languages must be sufficiently rich to represent natural languages
2. Learnability constraint: there must be a way for the child to learn these representations from the data available
3. Evolutionary constraint: it must not posit a rich, evolutionarily implausible language faculty

- An approach that potentially satisfies all three criteria.