# Distributional Learning

## Theoretical Linguistics and Formal Learning Theory

Alexander Clark

Manchester

6 November 2013

# Outline

# Outline

# American structuralism

## Structuralist tradition

- Bloomfield
- Rulon Wells
- Zellig Harris

# Harris, 1940

## Reviewing a nonstructuralist book

the value of the book is vitiated, especially for the layman, by a major short- coming. This is the neglect of the method of structural analysis, i.e. of organized synchronic description. As a result, many of the facts about languages are misconstrued, and linguistic theory is distorted. It is the chief purpose of this review to show that **an appreciation of linguistic structure is necessary for any interpretation of linguistics**, and that its neglect leads to undesirable results in practice.

Rejecting historical approach and its focus on written language.

# And its rejection . . .

## Chomsky, 1965

The only proposals that are explicit enough to support serious study are those that have been developed within taxonomic linguistics. It seems to have been demonstrated beyond reasonable doubt that quite apart from any questions of feasibility, methods of the sort that have been studied in taxonomic linguistics are intrinsically incapable of yielding the systems of grammatical knowledge that must be attributed to the speaker of a language.

# And its rejection . . .

## Chomsky, p.c.

From the 50s, there has seemed to me no hope in distributional procedures.

# Why was it rejected?

- Distributional learning was perceived as a discovery procedure
- No mathematically precise models (contra Harris)
- Problems of complexity unless the range of variation is finite
- A sequence of phrase markers – unlearnable because you only observe the last one (Katz and Postal, 1964)
- No way of dealing with structure-dependent movement
- No model of ambiguity or of syntactic structure

# Distinction

## Discovery procedure

Used by linguists to automatically generate a grammar
But if a grammar is a theory, why do we need to automatically generate it?

## Model of language acquisition

Rather than a linguist analyzing a corpus, we have a child processing the primary linguistic data

# Kulagina school

Structuralist linguistics died out in America after Chomsky.

## Oettinger, 1958

The latter paper (Kulagina 1957) has considerable expository merit, and it is clearer and more sensible than similar papers on set-theoretic concepts in language which have sprouted like ungainly weeds in the lawn of our information-retrieval literature. The work is along somewhat different lines, and of lesser extent but of caliber comparable to that of the excellent theoretical work of Chomsky in this country.

Solomon Marcus, Sestier, Kunze, . . .

# Direct psycholinguistic evidence

## Artificial Grammar Learning in infants

Saffran, Aslin, Newport (1996) ...

## Human simulation experiments

Gillette, J. et al. (1999), Gleitman (1990), ...

## Lexical acquisition experiments

Mintz, T. (2002), Childers and Tomasello (2001), ...

Children and adults do exploit distributional evidence.

# Computational experiments

## Natural language processing

Brown et al. (1992), Curran, J. (2003), . . .
Standard components of large NLP systems.

## CHILDES Experiments

Redington, Fitch, & Chater (1998), Mintz, T. (2003), . . .

These experiments show that rich evidence is available in reasonably sized natural corpora.

# Distributional learning

## Chomsky (1968/2006)

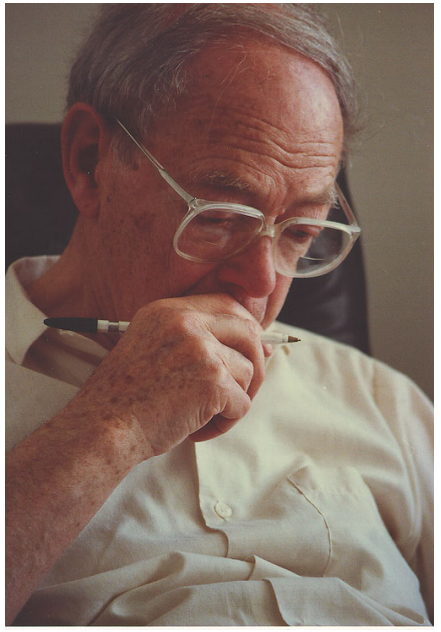
“The concept of "phrase structure grammar" was explicitly designed to express the richest system that could reasonable be expected to result from the application of Harris-type procedures to a corpus.”

# Outline

# Distributional Learning

## Zellig Harris (1949, 1951)

*Here as throughout these procedures X and Y are substitutable if for every utterance which includes X we can find (or gain native acceptance for) an utterance which is identical except for having Y in the place of X*



# Example

Is 'cat' substitutable for 'dog'?

- The cat is over there.
- I want a dog for Christmas.
- I want a Siamese cat for Christmas.
- Put a cat-flap in the door to the kitchen.
- An Alsatian is a breed of dog.
- He continues to dog my footsteps.
- I would rather have a dog than a cat as a pet.

# Example

Is 'cat' substitutable for 'dog'?

- The dog is over there.
- I want a cat for Christmas.
- I want a Siamese dog for Christmas.
- Put a dog-flap in the door to the kitchen.
- An Alsatian is a breed of cat.
- He continues to cat my footsteps.
- I would rather have a dog than a dog as a pet.
- I would rather have a cat than a dog as a pet.

# Empirical work on Distributional Learning

## Real corpora

- Sample is not just of grammatical sentences
- Also semantically well-formed
- Also “true” in some non-technical sense
- Empirical distribution is very complex

Distributional similarity in real corpora often reflects semantic relatedness

# Various notions of context

The word 'has' in the sentence: "If the candidate has an outstanding examination result"

## Local syntactic context

Immediately preceding and following word
(candidate, an)

## Wide bag-of-words context

Skip stop words
Set of words occurring in the same sentence/discourse.
{ candidate, examination, outstanding, result }

# Various notions of context

The word 'has' in the sentence: "If the candidate has an outstanding examination result"

## Local syntactic context

Immediately preceding and following word
(candidate, an)

## Wide bag-of-words context

Skip stop words
Set of words occurring in the same sentence/discourse.
{ candidate, examination, outstanding, result }

## Full context

"If the candidate □ an outstanding examination result"

# Example

## Distribution of “cat” in English

Infinite set of full contexts that ’cat’ can appear in :
“the □ is over there”
“I want a □ for Christmas”
. . .

- We can observe the distribution simply by looking at positive examples.
- We can see a similarity in distribution of “cat” and “dog”.
- Distributional learning is based on this idea.

# Distribution

## Full context

### Context (or *environment*)

A context is just a pair of strings $(l, r) \in \Sigma^* \times \Sigma^*$, that we write $l \Box r$

### Context substring relation

$l \Box r \odot u = lur$

Special context $\Box$

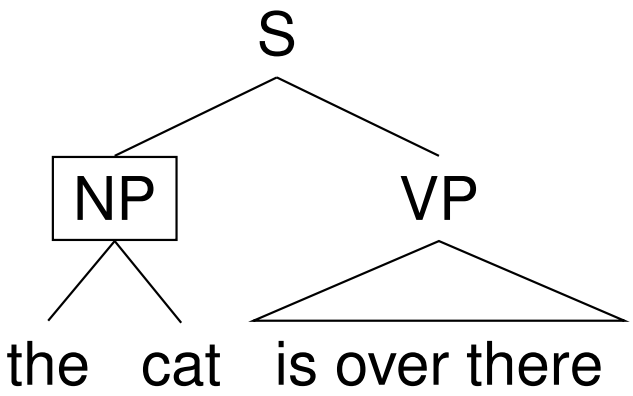
### Distribution of a string

Given a language $L$

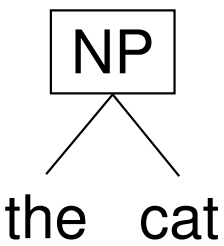
$C_L(u) = \{l \Box r | lur \in L\}$

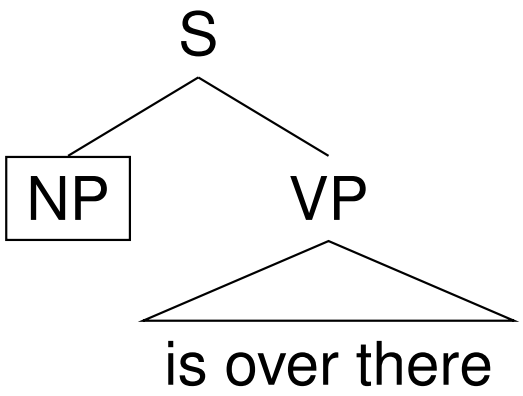
$\Box \in C_L(u)$ iff $u \in L$

# Trees

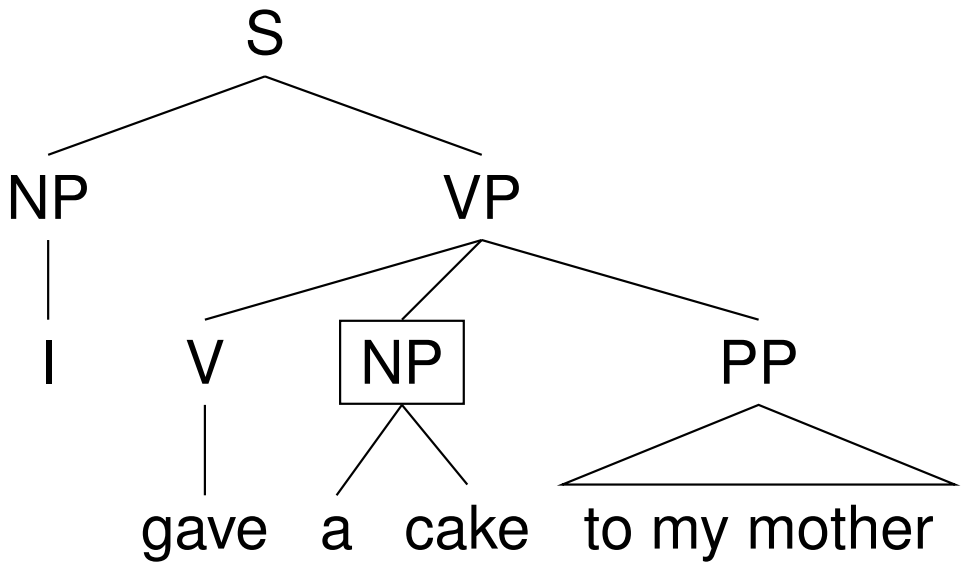


# Trees

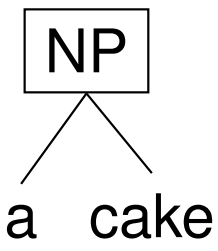


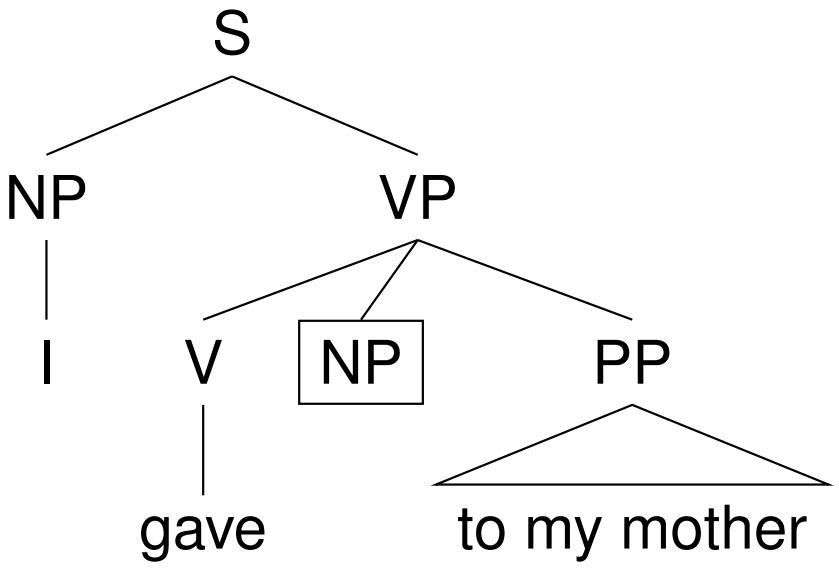



# Trees

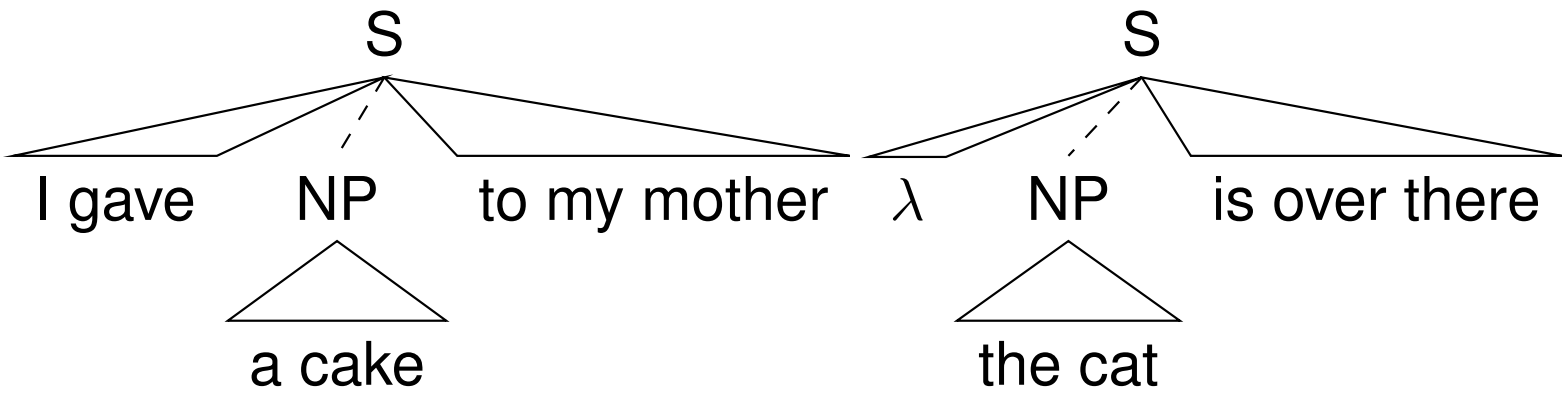


# Trees



# Contexts



# Contexts and yields of nonterminals

## Yield of a non-terminal

$Y_G(NP)$ is the set of all strings $w$ such that $NP \stackrel{*}{\Rightarrow} w$

$Y_G(NP) = \{$ the cat, the dog, some blue boxes $\ldots \}$

## Contexts of a non-terminal

$C_G(NP)$ is the set of all contexts $l\Box r$ such that $S \stackrel{*}{\Rightarrow} lNPr$

$C_G(NP) = \{ \Box$ is over there, I want $\Box$, Put it on $\Box \ldots \}$

Any string in $Y(NP)$ can occur in any context in $C(NP)$

# A difficult question

Suppose we have some string, say 'the cat', which is in $Y(NP)$

## Question

What is the relationship between the distribution of 'the cat'
$C_L$(the cat)
and the contexts or distribution of NP: $C_G(NP)$ ??

# Outline

# Old idea 1

## Chomsky review of Greenberg, 1959

let us say that two units A and B are substitutable$_1$ if there are expressions X and Y such that XAY and XBY are sentences of L.; substitutable$_2$ if whenever XAY is a sentence of L then so is XBY and whenever XBY is a sentence of L so is XAY (i.e. A and B are completely mutually substitutable). These are the simplest and most basic notions.

Problem:
we need substitutability$_2$ but what we observe is substitutability$_1$

# Old idea 2

## John Myhill, 1950 commenting on Bar-Hillel

I shall call a system *regular* if the following holds for all expressions $\mu, \nu$ and all wffs $\phi, \psi$ each of which contains an occurrence of $\nu$: If the result of writing $\mu$ for some occurrence of $\nu$ in $\phi$ is a wff, so is the result of writing $\mu$ for any occurrence of $\nu$ in $\psi$. Nearly all formal systems so far constructed are regular; ordinary word-languages are conspicuously not so.

## Clark and Eyraud, 2005/2007

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

# substitutable and reversible

## Clark and Eyraud, 2005

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

## Angluin, 1982

A language is *reversible* if $ur, vr, ur' \in L$ means that $vr' \in L$.

# A Bad Intuition

## One context in common in enough

- The cat died
- The dog died

So “cat” and “dog” are congruent

# A Bad Intuition

## One context in common in enough

- The cat died
- The dog died

So “cat” and “dog” are congruent

- He is an Englishman
- He is thin

So “thin” and “an Englishman” are congruent.

# Chomsky example

John is eager to please
John is easy to please

# Result

Clark and Eyraud, 2005/2007

## Polynomial result

The class of substitutable context free languages is polynomially identifiable in the limit from positive data only.

- Polynomial characteristic set
- Polynomial update time

Why the delay?

# A Simple Algorithm

## Non-technical description

- Given a sample of strings $W = \{w_1, \ldots, w_n\}$.
- Define a graph $G = \langle N, E \rangle$
  - $N$ is the set of all non-empty substrings (factors) of $W$.
  - $E = \{(u, v) | \exists l \Box r, lur \in W \wedge lvr \in W\}$.
- Define a grammar in Chomsky normal form
  - The set of non-terminals is the set of components of the graph $G$
  - Have productions for: $[a] \to a$
  - Add rules for non terminals: $[w] \to [u][v]$ iff $[w] = [uv]$.

# Simple linguistically motivated example

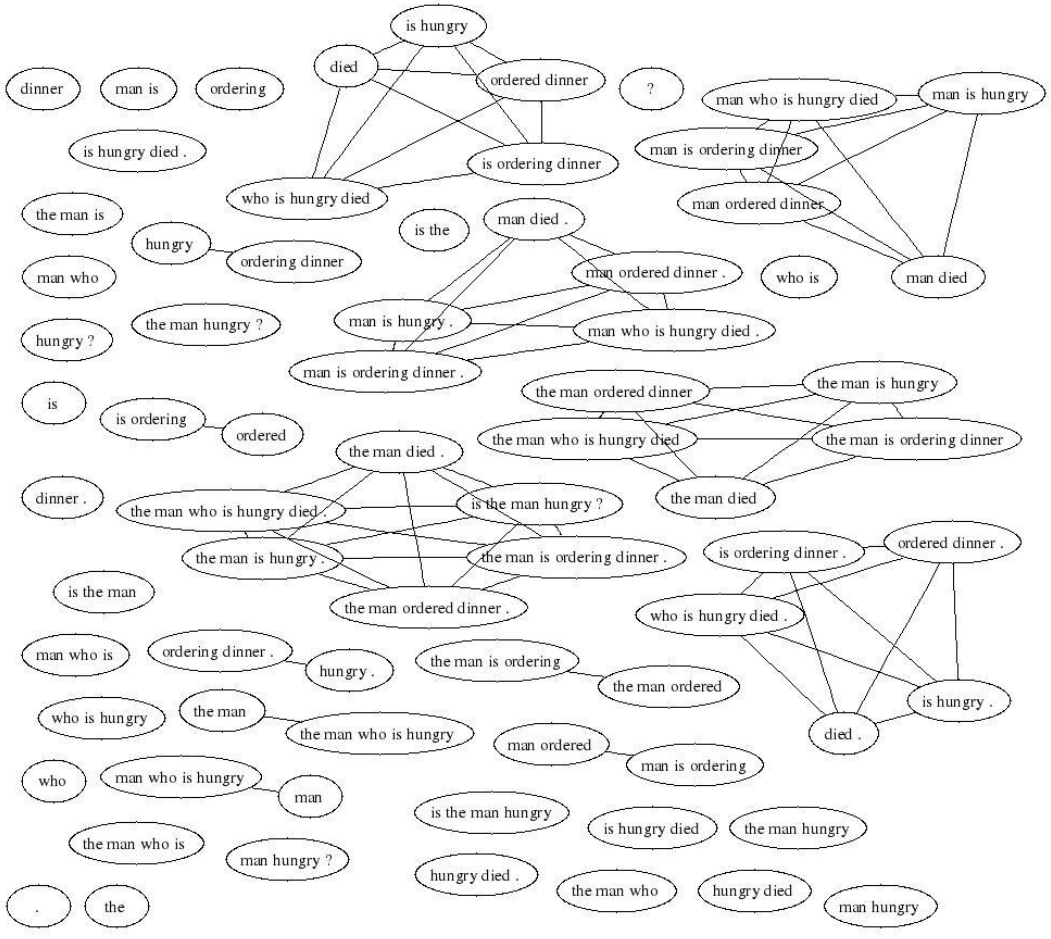
the man who is hungry died .
the man ordered dinner .
the man died .
the man is hungry .
is the man hungry ?
the man is ordering dinner .

---

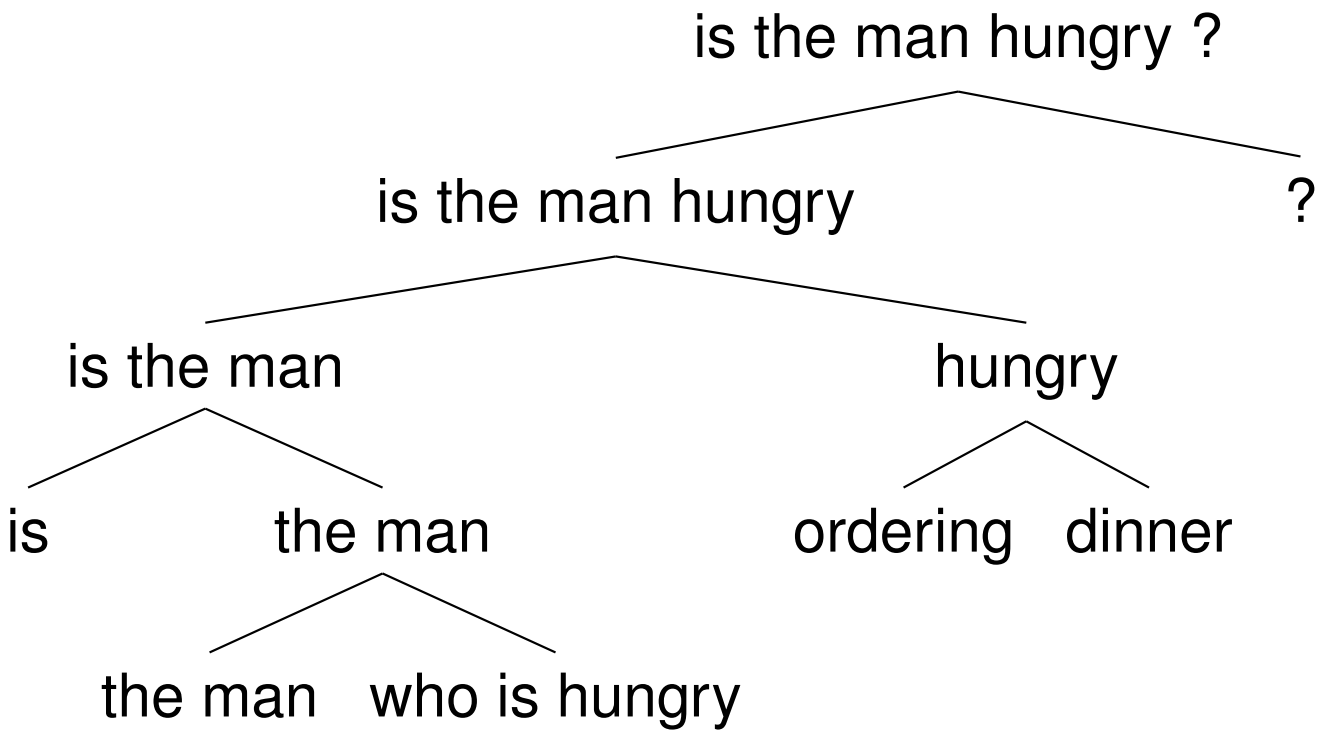
is the man who is hungry ordering dinner ?
$*$is the man who hungry is ordering dinner ?

# Substitution graph

## Auxiliary fronting example



# Tree


is the man hungry ?
is the man hungry
?
is the man
hungry
is
the man
ordering
dinner
the man
who is hungry

# Counterexample

Berwick, Coen and Niyogi, p.c

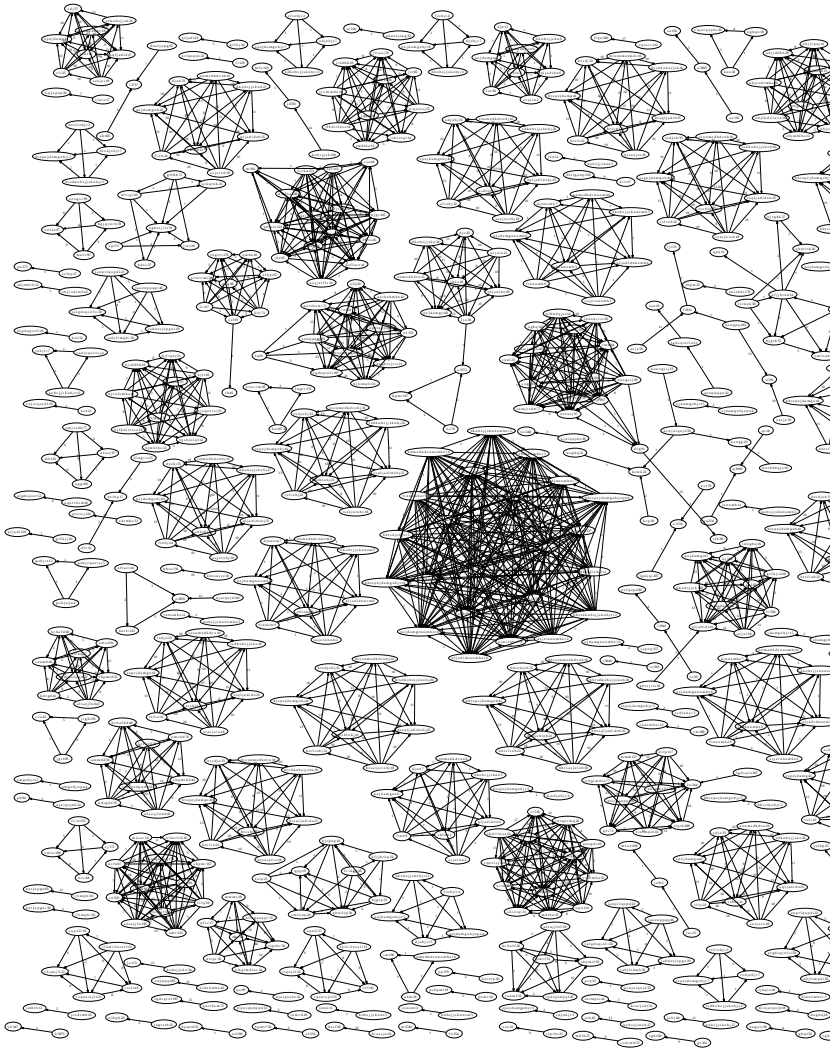
is (bob) well ?
is (the man john) well ?
does he think (well) ?
does he think (hitting is nice) ?
is (bob) (well) ?

---

is the man john hitting is nice ?

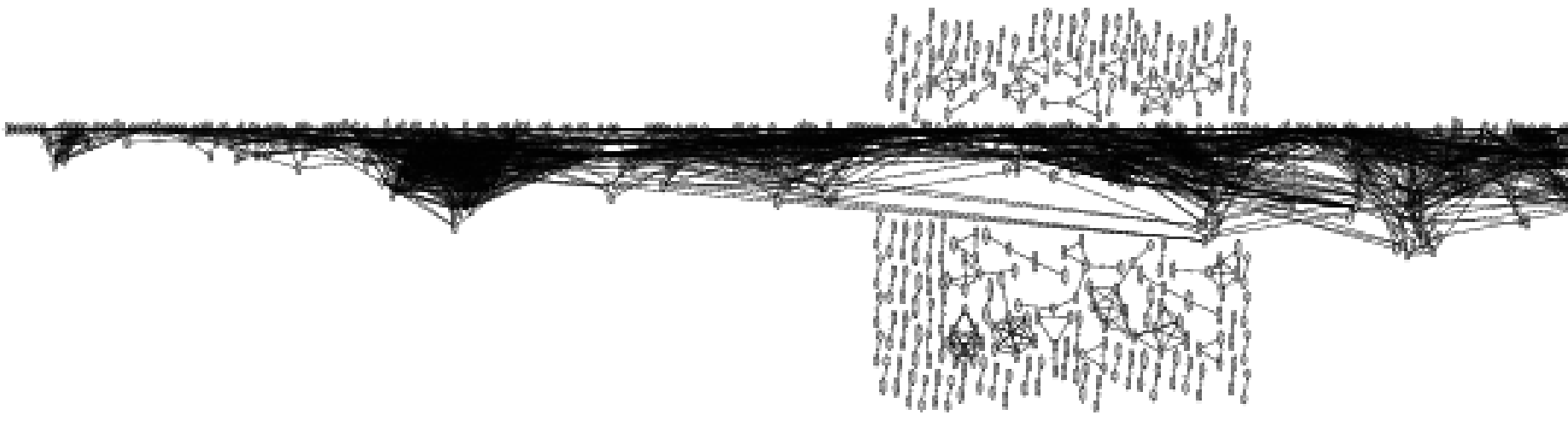
# Substitution graph

## Learnable example



# Substitution graph

## Unlearnable languages



# Substitutable languages

- Some very basic languages are not substitutable:
  - $L = \{a, aa\}$
  - $L = \{a^n b^n | n > 0\}$
  - Dyck language
- The very strict requirement for contexts to be disjoint is unrealistic.
- The test for congruence is way too weak.
- If we move to a probabilistic learning approach, or queries we can have a better test
- If the test is good, the hypothesis will never overgenerate

# Outline

# Berwick, Chomsky critique

[Berwick et al.(2011)Berwick, Pietroski, Yankama, and Chomsky]

> *Put another way, language acquisition is not merely a matter of acquiring a capacity to associate word strings with interpretations. Much less is it a mere process of acquiring a (weak generative) capacity to produce just the valid word strings of a language. Idealizing, one can say that each child acquires a procedure that generates boundlessly many meaningful expressions, and that a single string of words can correspond to more than one expression.*

# Two options

- Learn from sound/meaning pairs
- Learn trees just from strings

# Alternative model

## Two mathematically reasonable models

| | Inputs | Outputs |
|---|---|---|
| Weak | strings | strings |
| Weak semantic | strings + meanings | strings + meanings |

# Alternative model

## Two mathematically reasonable models

| | Inputs | Outputs |
|---|---|---|
| Weak | strings | strings |
| Weak semantic | strings + meanings | strings + meanings |

## A mathematically unreasonable model

[Wexler and Culicover(1980)]

| | Inputs | Outputs |
|---|---|---|
| Strong learning | strings | strings + trees |

# Structural descriptions as derivation trees of CFGs

## Target class of grammars

$\mathcal{G}$ is some set of context-free grammars.
Pick some grammar $G_* \in \mathcal{G}$

## Weak learning

We receive examples $w_1, \ldots, w_n, \ldots$
We produce a series of hypotheses $G_1, \ldots, G_n, \ldots$
We want $G_n$ to converge to some grammar $\hat{G}$ such that $L(\hat{G}) = L(G_*)$

# Structural descriptions as derivation trees of CFGs

## Target class of grammars

$\mathcal{G}$ is some set of context-free grammars.
Pick some grammar $G_* \in \mathcal{G}$

## Strong learning

We receive examples $w_1, \ldots, w_n, \ldots$
We produce a series of hypotheses $G_1, \ldots, G_n, \ldots$
We want $G_n$ to converge to some grammar $\hat{G}$ such that $\hat{G} \equiv G_*$

# Equivalence

**Equality** $\hat{G} = G_*$

**Isomorphism** $\hat{G} \equiv G_*$: a bijection between the nonterminals so that the two grammars have the same productions.

**Strong equivalence** Isomorphic trees

**Structural equivalence** Same unlabelled parse trees [Paull and Unger(1968)]

**Weak equivalence** $L(\hat{G}) = L(G_*)$

# Redundancy

## Claim

If we have two grammars $G_1, G_2 \in \mathcal{G}$ which are:

1. Weakly equivalent: $L(G_1) = L(G_2)$
2. Not strongly equivalent: $G_1 \not\equiv G_2$

then we can't strong-learn $\mathcal{G}$.

# Redundancy

## Claim

If we have two grammars $G_1, G_2 \in \mathcal{G}$ which are:

1. Weakly equivalent: $L(G_1) = L(G_2)$
2. Not strongly equivalent: $G_1 \not\equiv G_2$

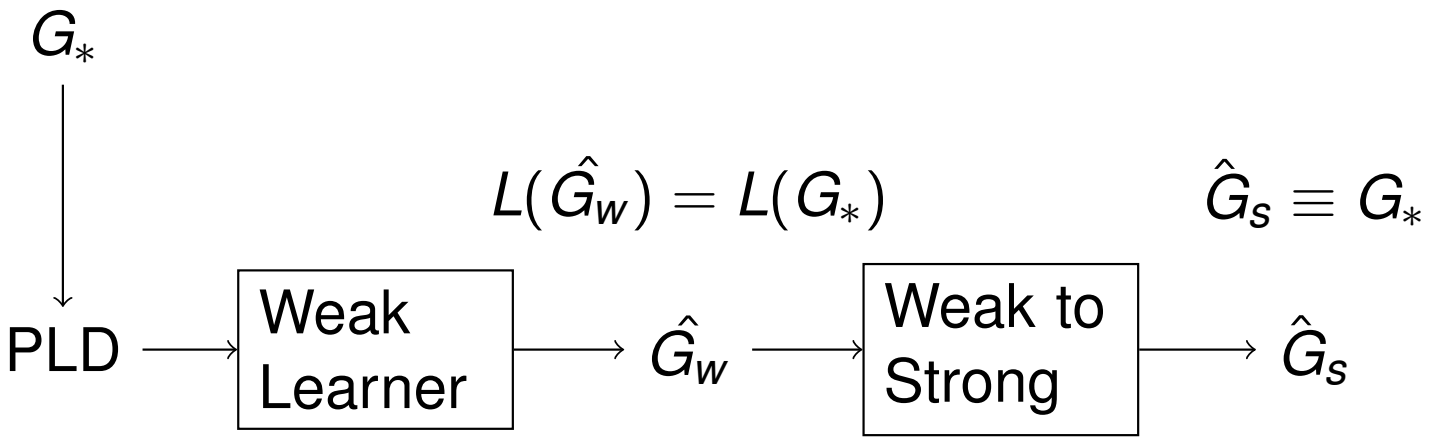
then we can't strong-learn $\mathcal{G}$.

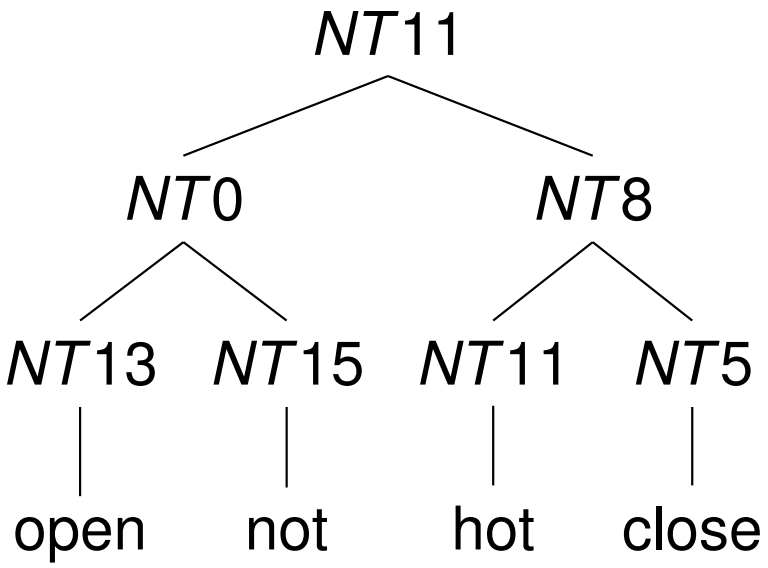
## Canonical grammars

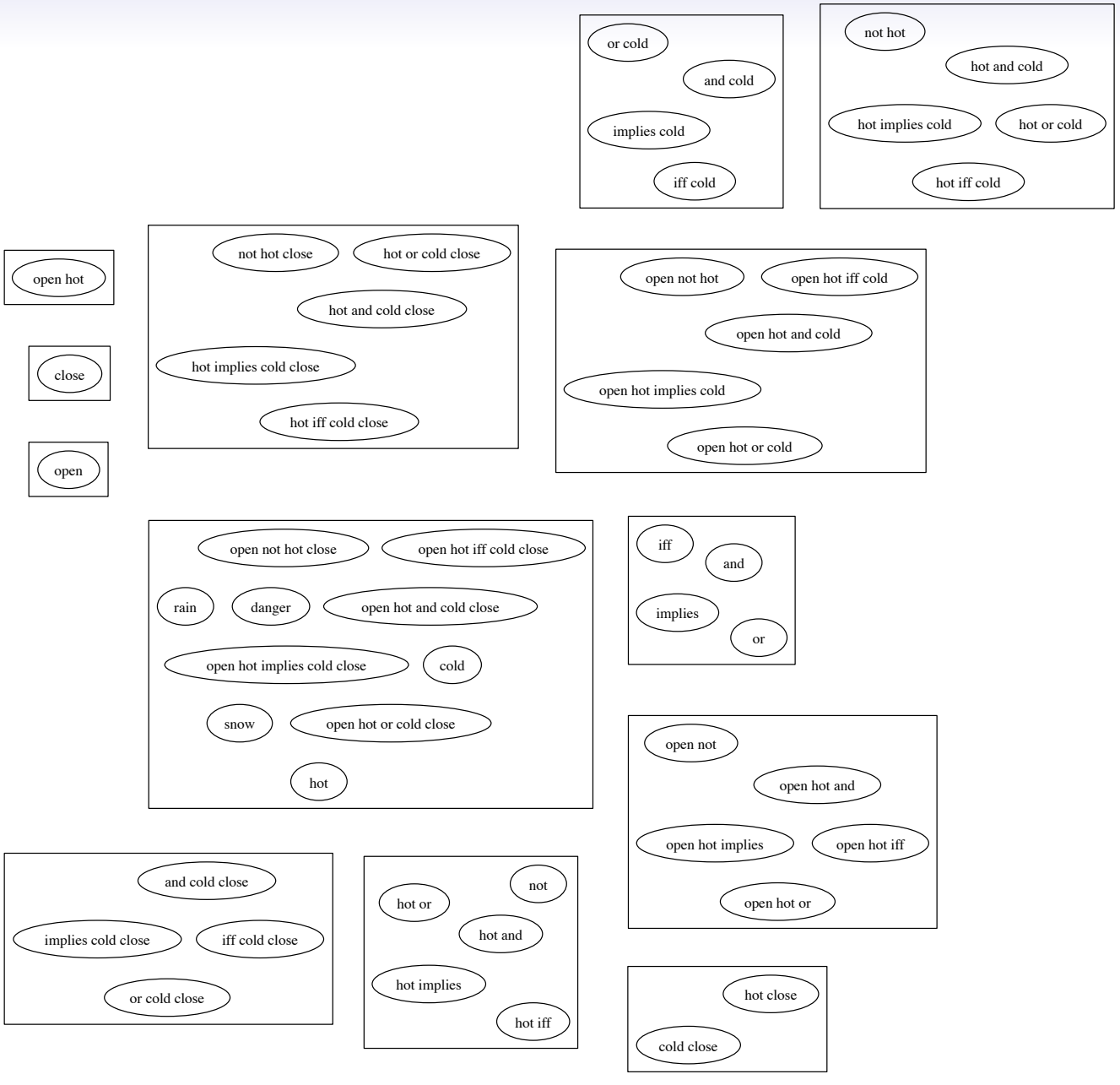
We can only have one 'distinct' grammar for each language in $\mathcal{G}$.

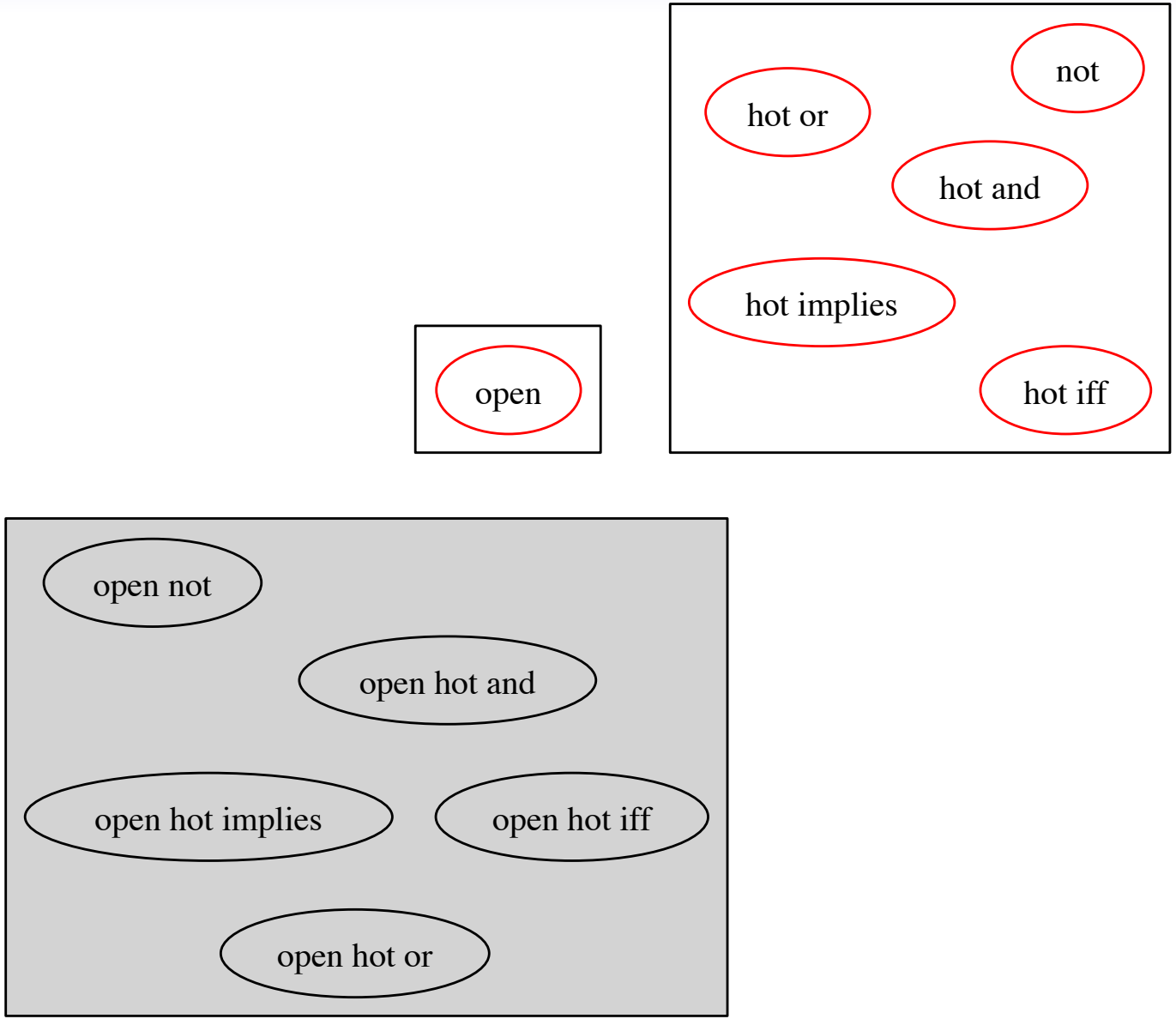
Contrast with : "a language does not uniquely determine the grammar that generates it"[Lewis(1975)]

# Weak to Strong learning

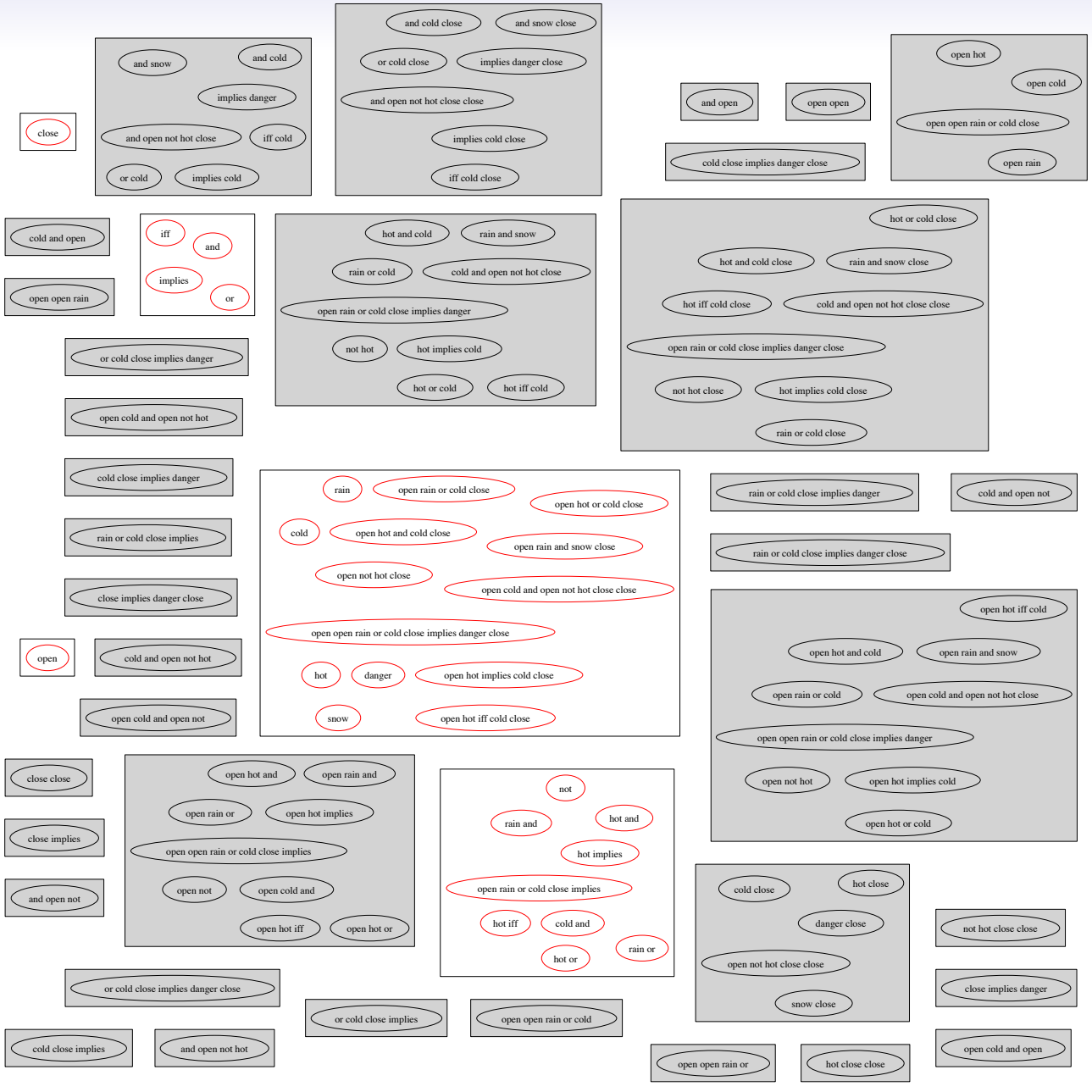



NT11
NT0
NT8
NT13
NT15
NT11
NT5
open
not
hot
close


or cold
and cold
implies cold
iff cold
not hot
hot and cold
hot implies cold
hot or cold
hot iff cold
open hot
close
open
not hot close
hot or cold close
hot and cold close
hot implies cold close
hot iff cold close
open not hot
open hot iff cold
open hot and cold
open hot implies cold
open hot or cold
open not hot close
open hot iff cold close
rain
danger
open hot and cold close
open hot implies cold close
cold
snow
open hot or cold close
hot
iff
and
implies
or
open not
open hot and
open hot implies
open hot iff
open hot or
and cold close
implies cold close
iff cold close
or cold close
not
hot or
hot and
hot implies
hot iff
hot close
cold close


not
hot or
hot and
hot implies
hot iff
open
open not
open hot and
open hot implies
open hot iff
open hot or


or cold
and cold
implies cold
iff cold
not hot
hot and cold
hot implies cold
hot or cold
hot iff cold
open hot
close
open
not hot close
hot or cold close
hot and cold close
hot implies cold close
hot iff cold close
open not hot
open hot iff cold
open hot and cold
open hot implies cold
open hot or cold
open not hot close
open hot iff cold close
rain
danger
open hot and cold close
open hot implies cold close
cold
snow
open hot or cold close
hot
iff
and
implies
or
open not
open hot and
open hot implies
open hot iff
open hot or
and cold close
implies cold close
iff cold close
or cold close
not
hot or
hot and
hot implies
hot iff
hot close
cold close


close
and snow
and cold
implies danger
and open not hot close
iff cold
or cold
implies cold
and cold close
and snow close
or cold close
implies danger close
and open not hot close close
implies cold close
iff cold close
and open
open open
cold close implies danger close
open hot
open cold
open open rain or cold close
open rain
cold and open
open open rain
iff
and
implies
or
hot and cold
rain and snow
rain or cold
cold and open not hot close
open rain or cold close implies danger
not hot
hot implies cold
hot or cold
hot iff cold
hot or cold close
hot and cold close
rain and snow close
hot iff cold close
cold and open not hot close close
open rain or cold close implies danger close
not hot close
hot implies cold close
rain or cold close
or cold close implies danger
open cold and open not hot
cold close implies danger
rain or cold close implies
close implies danger close
open
cold and open not hot
open cold and open not
rain
open rain or cold close
open hot or cold close
cold
open hot and cold close
open rain and snow close
open not hot close
open cold and open not hot close close
open open rain or cold close implies danger close
hot
danger
open hot implies cold close
snow
open hot iff cold close
rain or cold close implies danger
cold and open not
rain or cold close implies danger close
open hot iff cold
open hot and cold
open rain and snow
open rain or cold
open cold and open not hot close
open open rain or cold close implies danger
open not hot
open hot implies cold
open hot or cold
close close
close implies
and open not
open hot and
open rain and
open rain or
open hot implies
open open rain or cold close implies
open not
open cold and
open hot iff
open hot or
not
rain and
hot and
hot implies
open rain or cold close implies
hot iff
cold and
hot or
rain or
cold close
hot close
danger close
open not hot close close
snow close
not hot close close
close implies danger
open cold and open
or cold close implies danger close
cold close implies
and open not hot
or cold close implies
open open rain or cold
open open rain or
hot close close

# Definition

## Definition

A congruence class $X$ is composite if there are two congruence classes $Y, Z$ such that $X = YZ$.
(and neither $Y$ nor $Z$ is the class $[\lambda]$

## Definition

A congruence class $X$ is prime if it is not composite.

The whole is greater than the sum of the parts

## Fundamental theorem of substitutable languages

Every congruence class $Q$ can be uniquely represented as a sequence of primes such that $Q = P_1 \dots P_n$

## Fundamental theorem of substitutable languages

Every congruence class $Q$ can be uniquely represented as a sequence of primes such that $Q = P_1 \dots P_n$

## Intuition

If $X = YZ$, and we have a rule $P \to QXR$, then we can change it to $P \to QYZR$

# Restriction

- We only consider substitutable languages which have a finite number of primes.
- We define nonterminals only for these primes.

| Label | Examples |
|---|---|
| P | rain, cold, open rain and cold close |
| O | open |
| C | close |
| B | and, or, ... |
| N | not, hot or, cold and ... |

# Productions

We need non-binary rules.

## Correct productions

$P_0 \to P_1 \dots P_k$
where $P_0 \supsetneq P_1 \dots P_k$

- $N \to PB$
- $P \to ONSC$
- Not $P \to OPBPC$ – correct but too long.
- Not $S \to ONONSCC$

If there are $n$ primes then there are at most $n^2$ valid productions.

# A Strong Learning Result

## Class of grammars

$\mathcal{G}_{sc}$ is the class of canonical grammars for all substitutable languages with a finite number of primes.

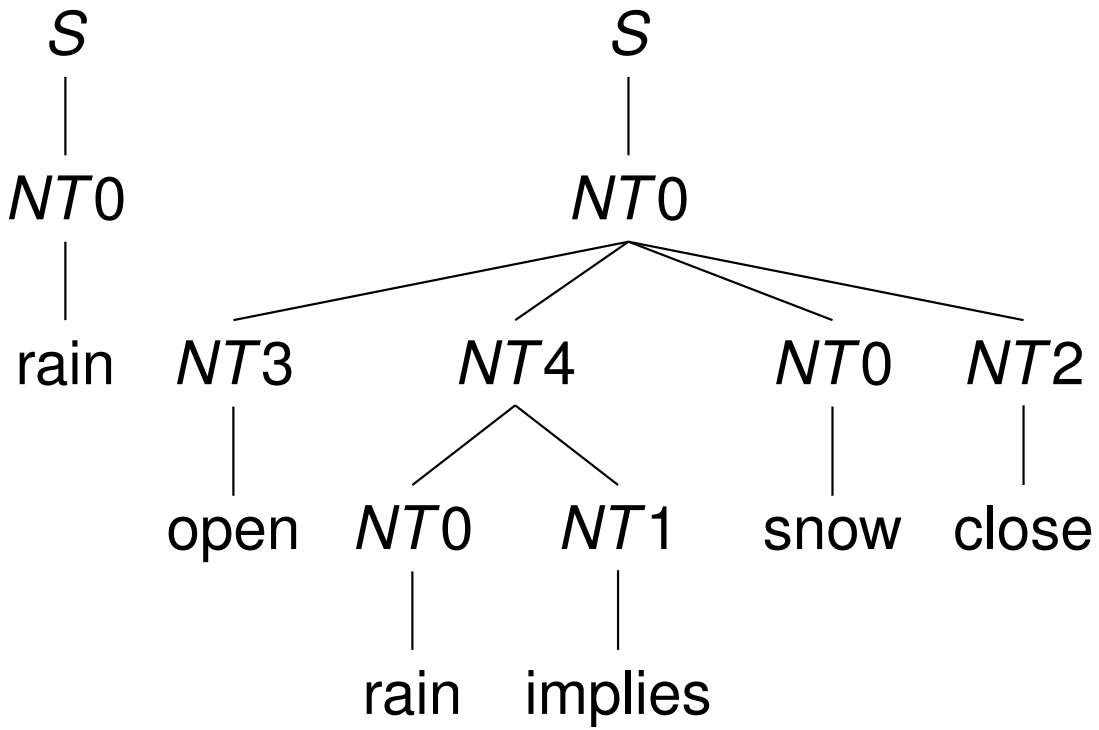
## Theorem

There is an algorithm which learns $\mathcal{G}_{sc}$

- From positive examples
- Identification in the limit convergence
- Strongly
- Using polynomial time and data

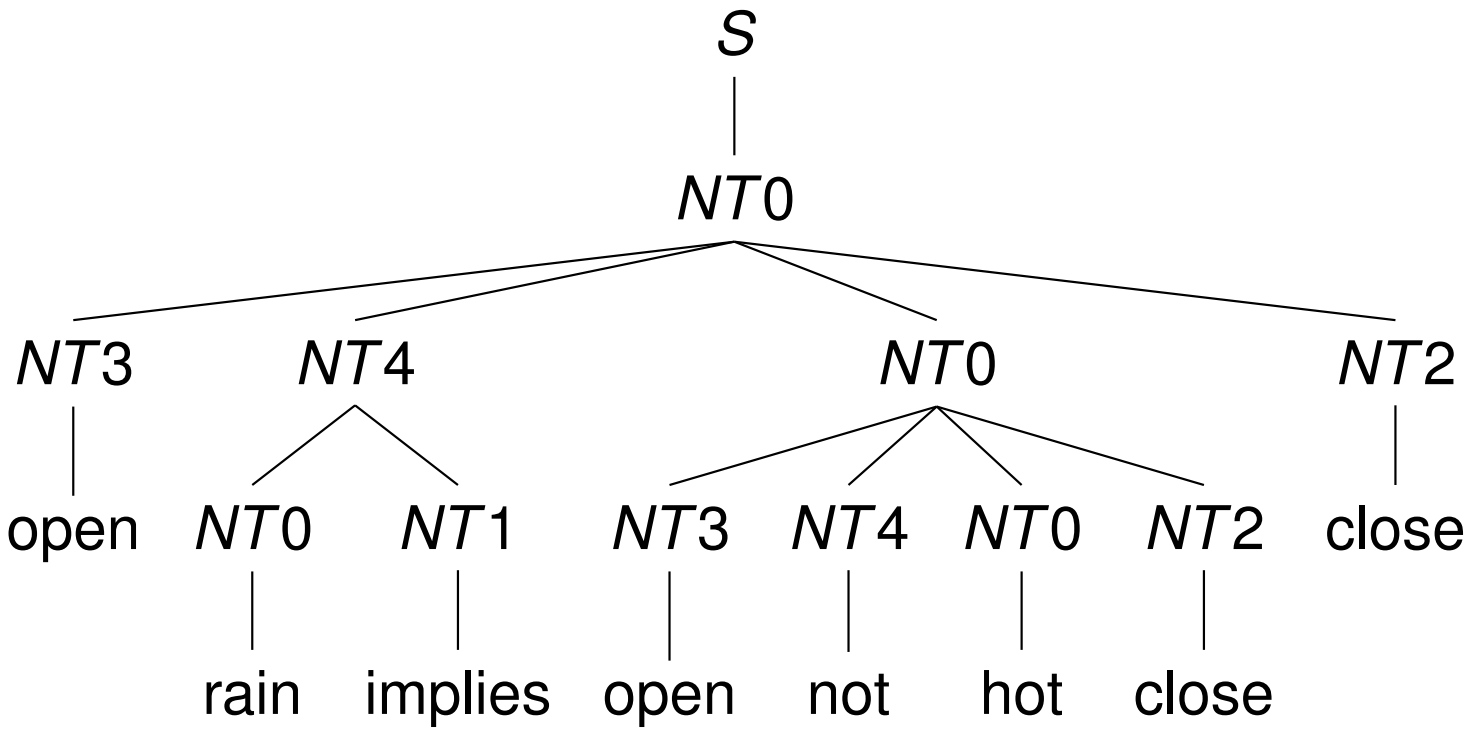
# Running example

(verbatim output from implementation)



# Running example

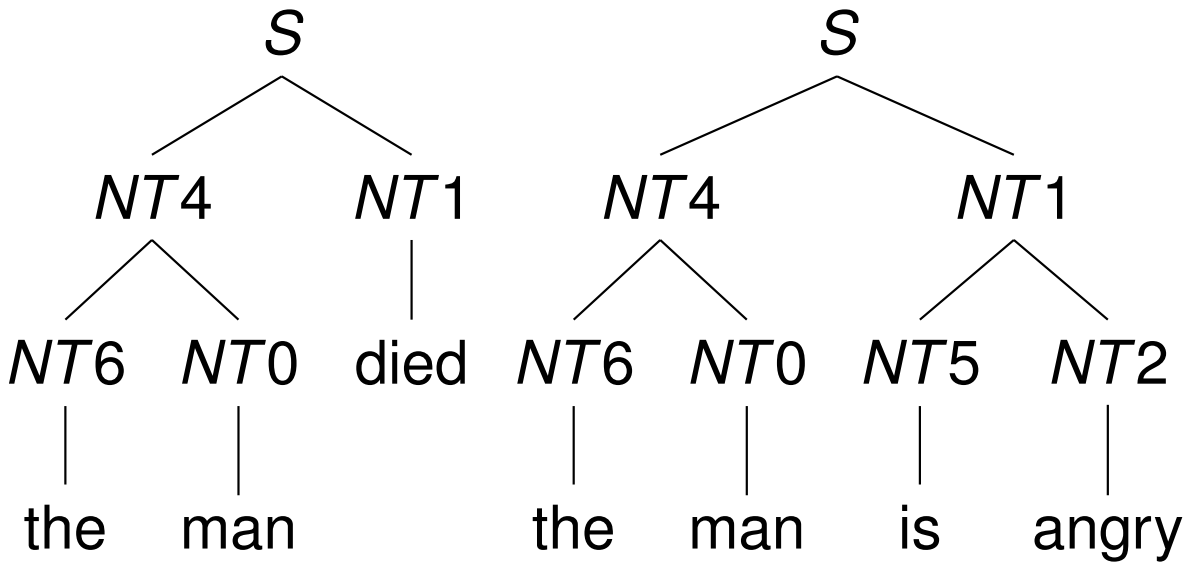
(verbatim output from implementation)



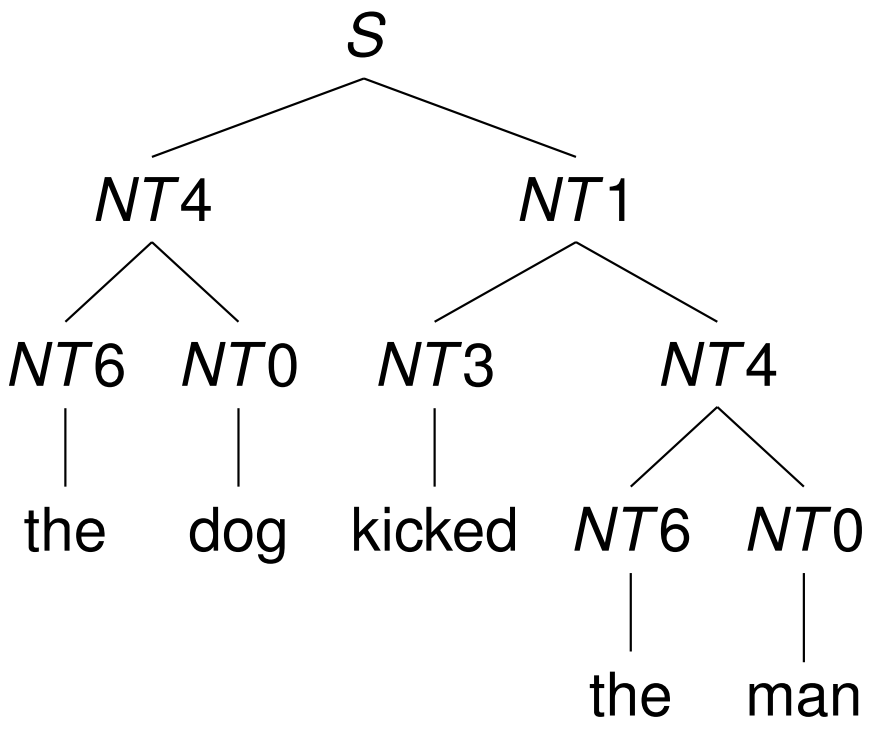
# Example 1

1. the man died
2. the dog died
3. the man is angry
4. it died
5. he died
6. the man kicked the dog

# Example 1


S
NT4
NT1
NT6
NT0
died
the
man
S
NT4
NT1
NT6
NT0
NT5
NT2
the
man
is
angry

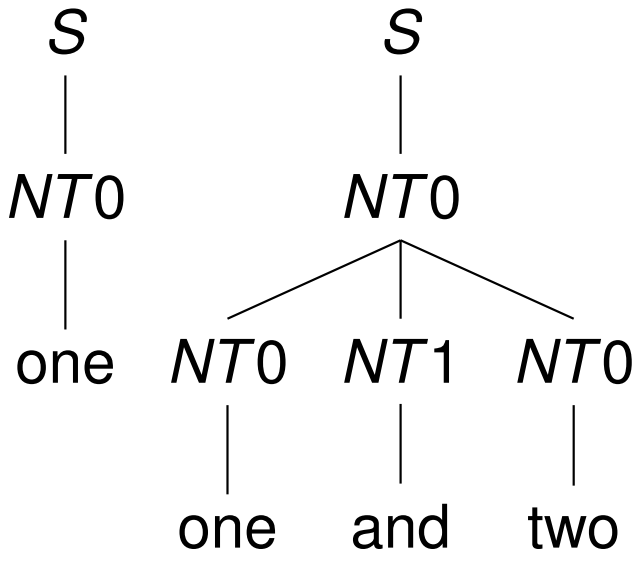
# Example 1



# Example 2

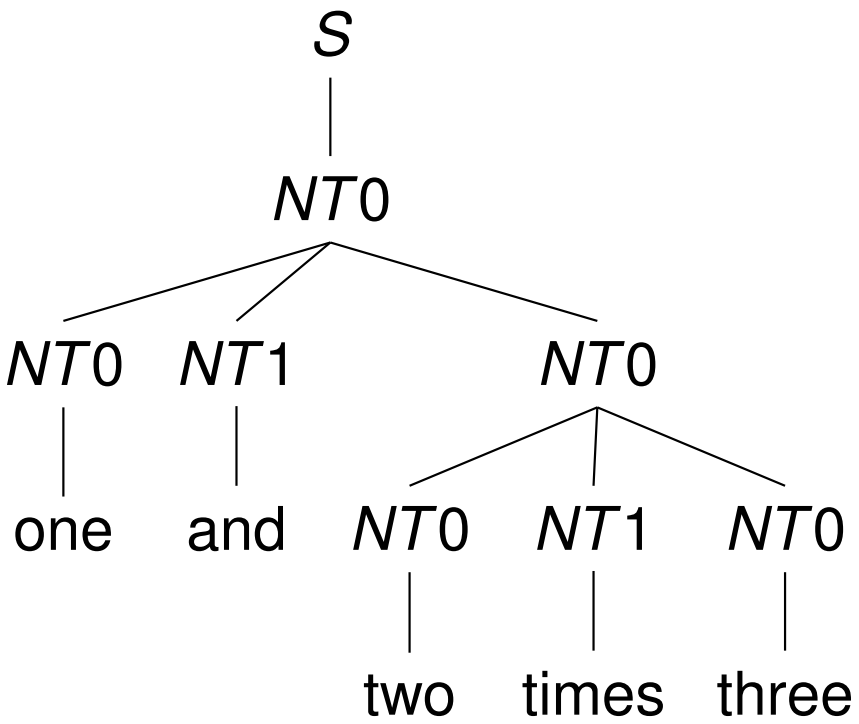
1. one
2. two
3. three
4. one and two
5. two times three
6. one and two times three

# Example 2



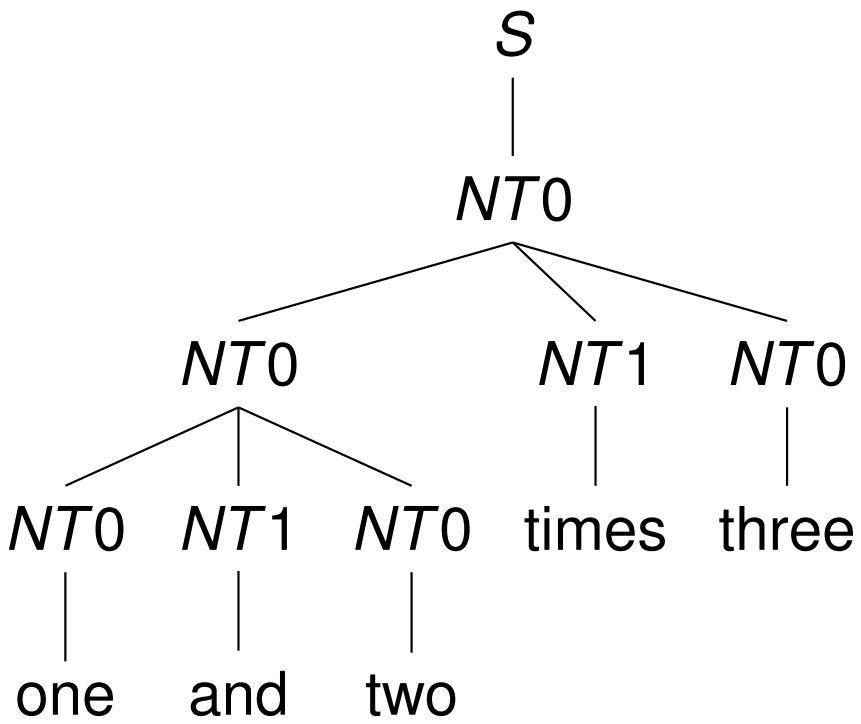
# Example 2

## Three trees for this string



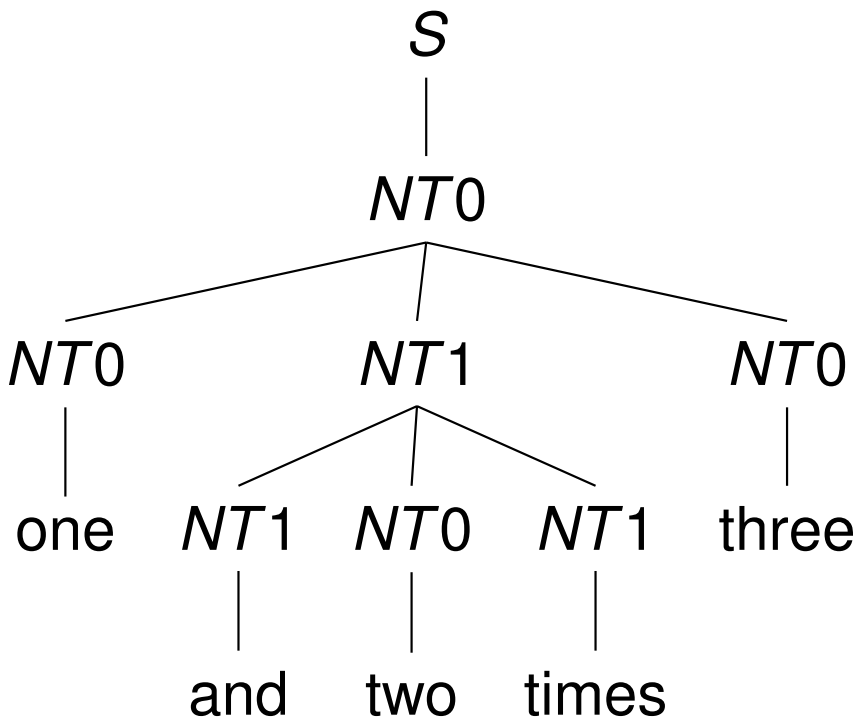
# Example 2

## Three trees for this string



# Example 2

## Three trees for this string



# Result

## Theorem: [Clark(2013b)]

This algorithm can learn all substituable languages with a finite number of primes:

- Rapidly, efficiently
- Strongly
- From positive data alone

# Outline

# Limitations of congruential approach

- Not context sensitive
- Classes are too small: need many non-terminals
- Fails to capture generalisations
- No notion of feature
- Hard to tell very similar classes apart
- Lattice based approaches (Clark FG 2009, CoNLL 2010, ICGI 2010)

# Example

"cat" in the sentence "There is a cat over there"

## Smallest class

Set of all strings that can be substituted for "cat" in all contexts
Possibly only "cat"

## Largest class

The set of all strings that can occur in a single context
"There is a □ over there"

- cat
- dog
- large cat
- large cat near here and a small dog

# Multiple contexts

## Better

The set of all strings that can occur in **both** contexts

1. “There is a □ over there”
2. “The □ just ran out”

# Multiple contexts

## Better

The set of all strings that can occur in **both** contexts

1. "There is a □ over there"
2. "The □ just ran out"

## Contexts are often "ambiguous"

"He is □"

- boring
- a lawyer
- away on holiday

# Finite context property

## Example

I put the book on the table.

## A context for the NP

I put the book on □.

# Finite context property

## Example

I put the book on the table.

## A context for the NP

I put the book on □.

- I put the book on the table.
- I put the book on the floor.
- I put the book on the table and then went outside.

# Finite context property

## Example

I put the book over there.

## A context for the NP

I put □ over there.

# Finite context property

## Example

I put the book over there.

## A context for the NP

I put □ over there.

- I put the book over there.
- I put the wine over there.
- I put the book in the bin and then put the wine over there.

# Finite context property

## Two contexts for the NP

I put □ over there.

AND

I put the book on □.

- I put the table over there.
- I put the book on the table.

# Finite context property

## Two contexts for the NP

I put □ over there.

AND

I put the book on □.

- I put the book in the bin and then put the wine over there.
- *I put the book on the book in the bin and then put the wine.

# Finite context property

## Two contexts for the NP

I put □ over there.

AND

I put the book on □.

- *I put the table and then went outside over there.
- I put the book on the table and then went outside.

# Notation

Suppose $C$ is a set of contexts. We write

$$C^{\triangleleft}$$

for the set of strings that can occur in all of the contexts of $C$.

# Notation

Suppose $C$ is a set of contexts. We write

$$C^{\triangleleft}$$

for the set of strings that can occur in all of the contexts of $C$.

## Finite Context Property

A grammar has the FCP if for each category (nonterminal) we can find a finite set of contexts that picks out the strings in that category.

$$C^{\triangleleft} = \mathcal{L}(G, N)$$

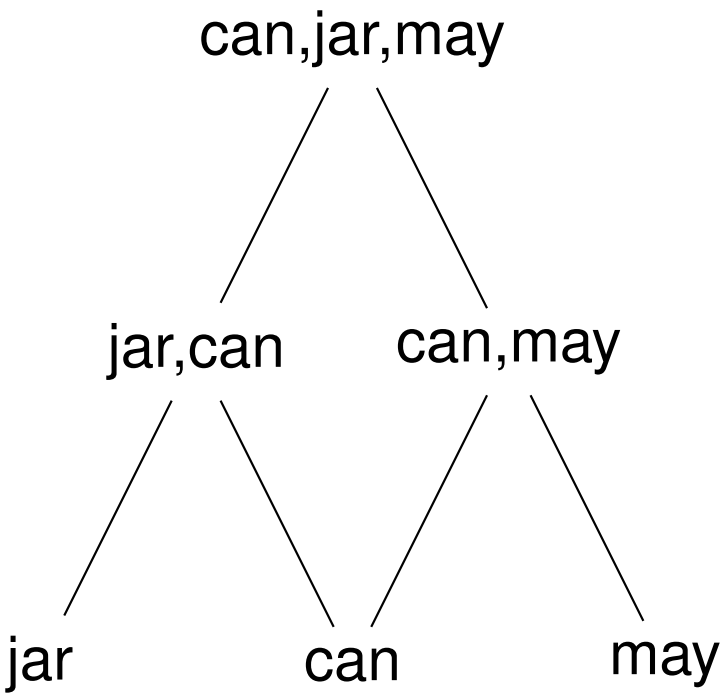
# Ambiguity in congruential models

## Examples

- Can I have a can of beans?
- May I have a jar of beans?

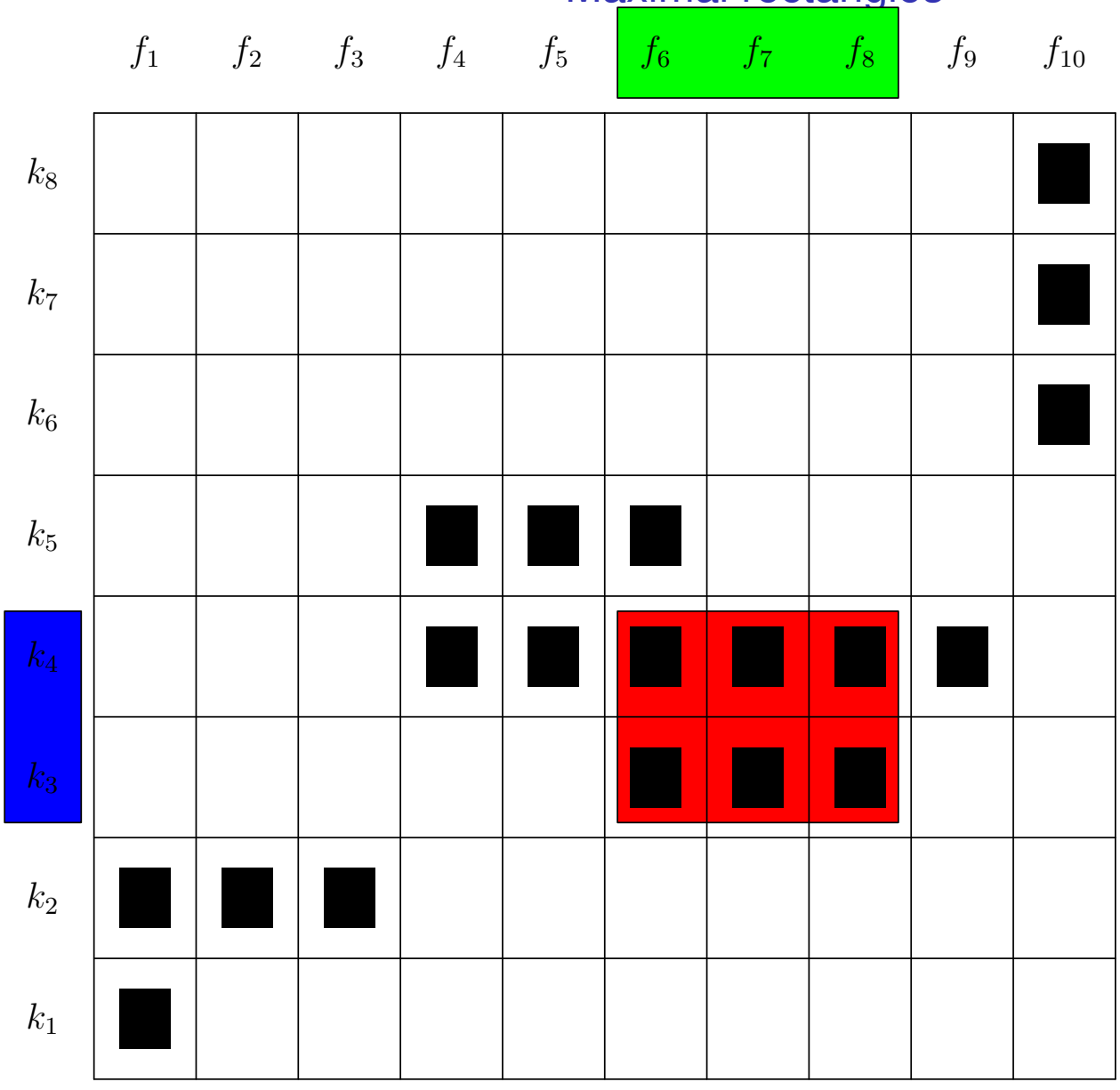
- “can” and “may” are different distributionally
- “can” and “jar” are different distributionally
- The structural descriptions are thus completely distinct

# Ambiguity in lattice models



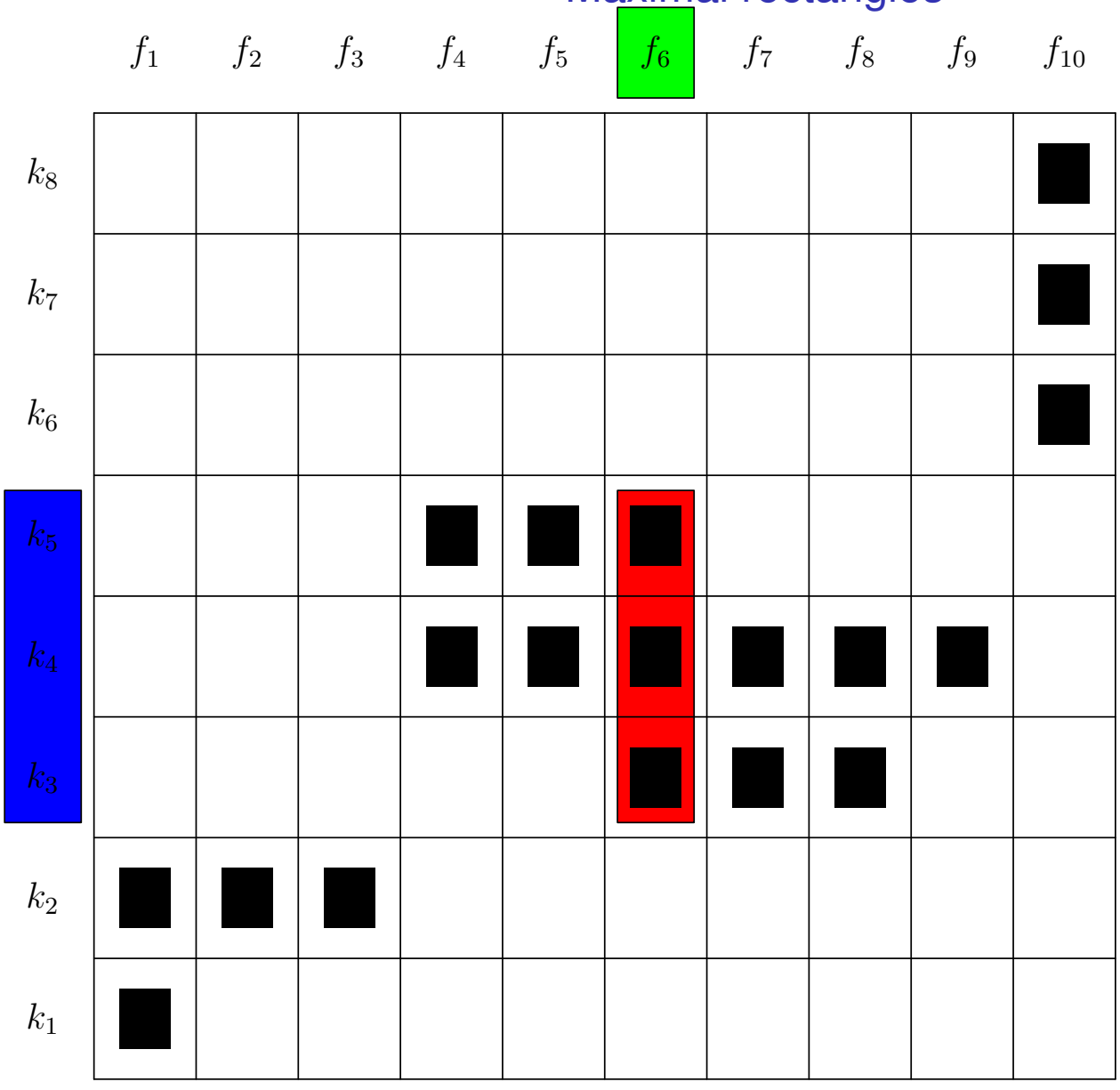
# Concepts

## Maximal rectangles



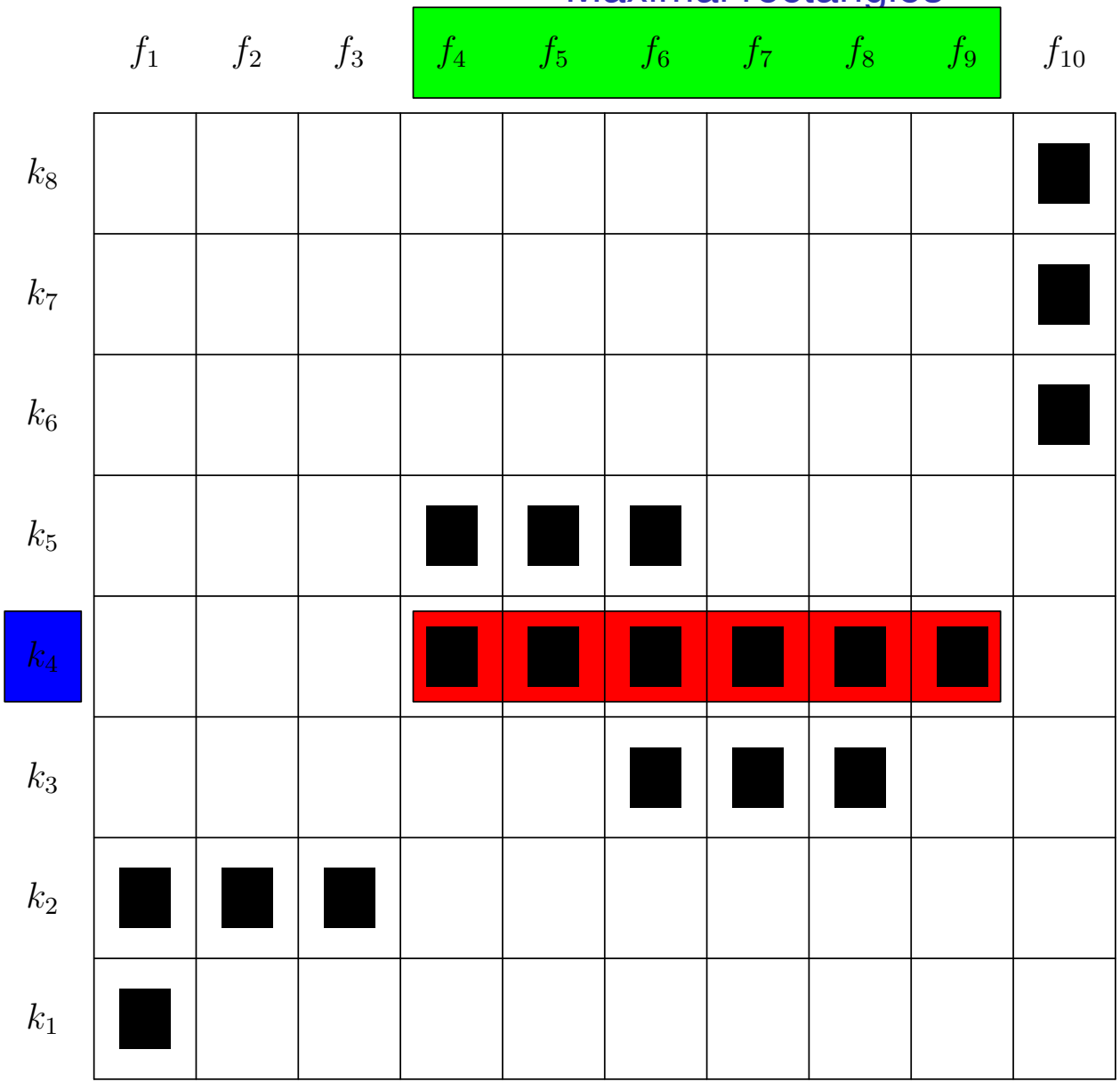
# Concepts

## Maximal rectangles



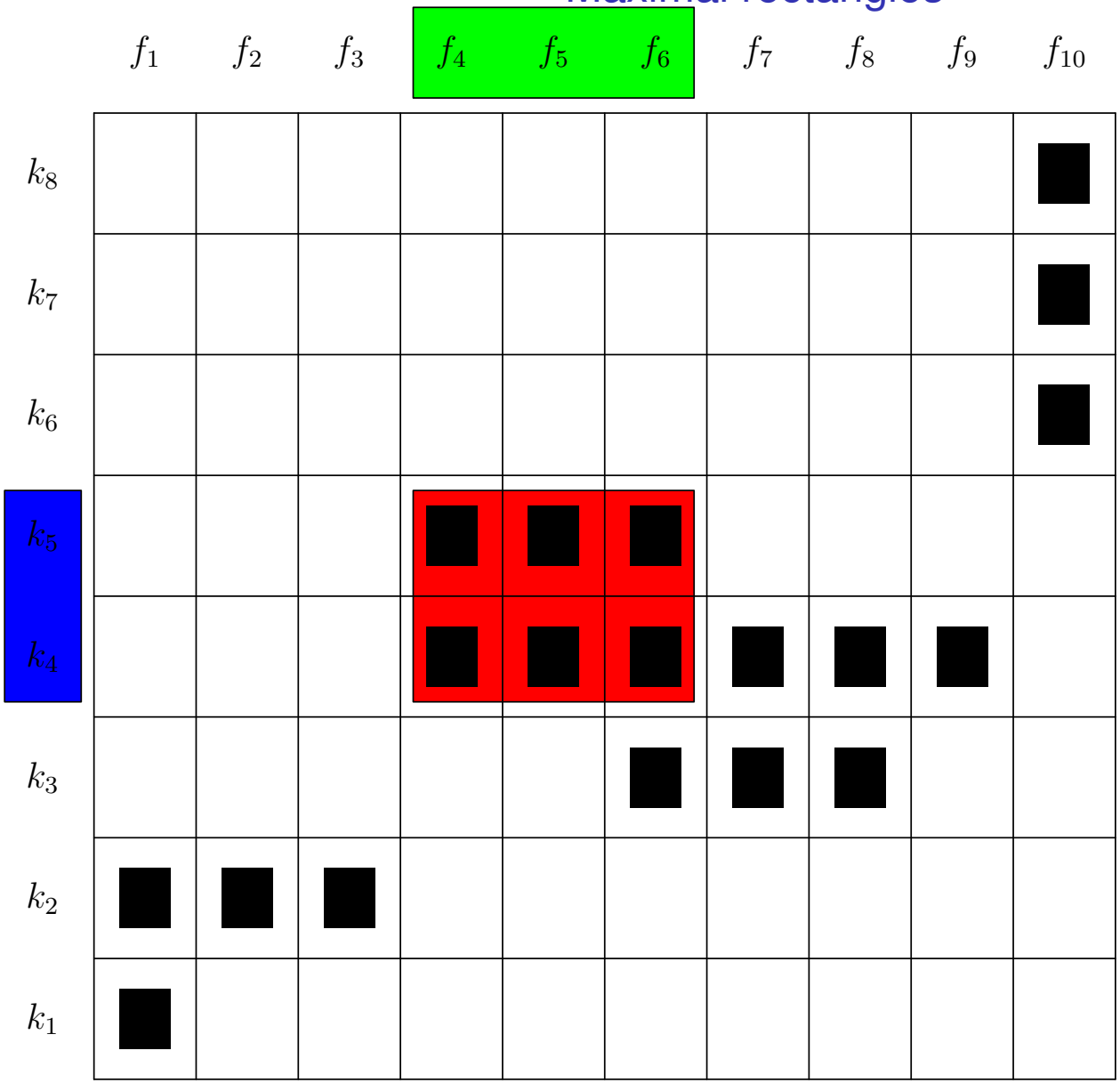
# Concepts

## Maximal rectangles



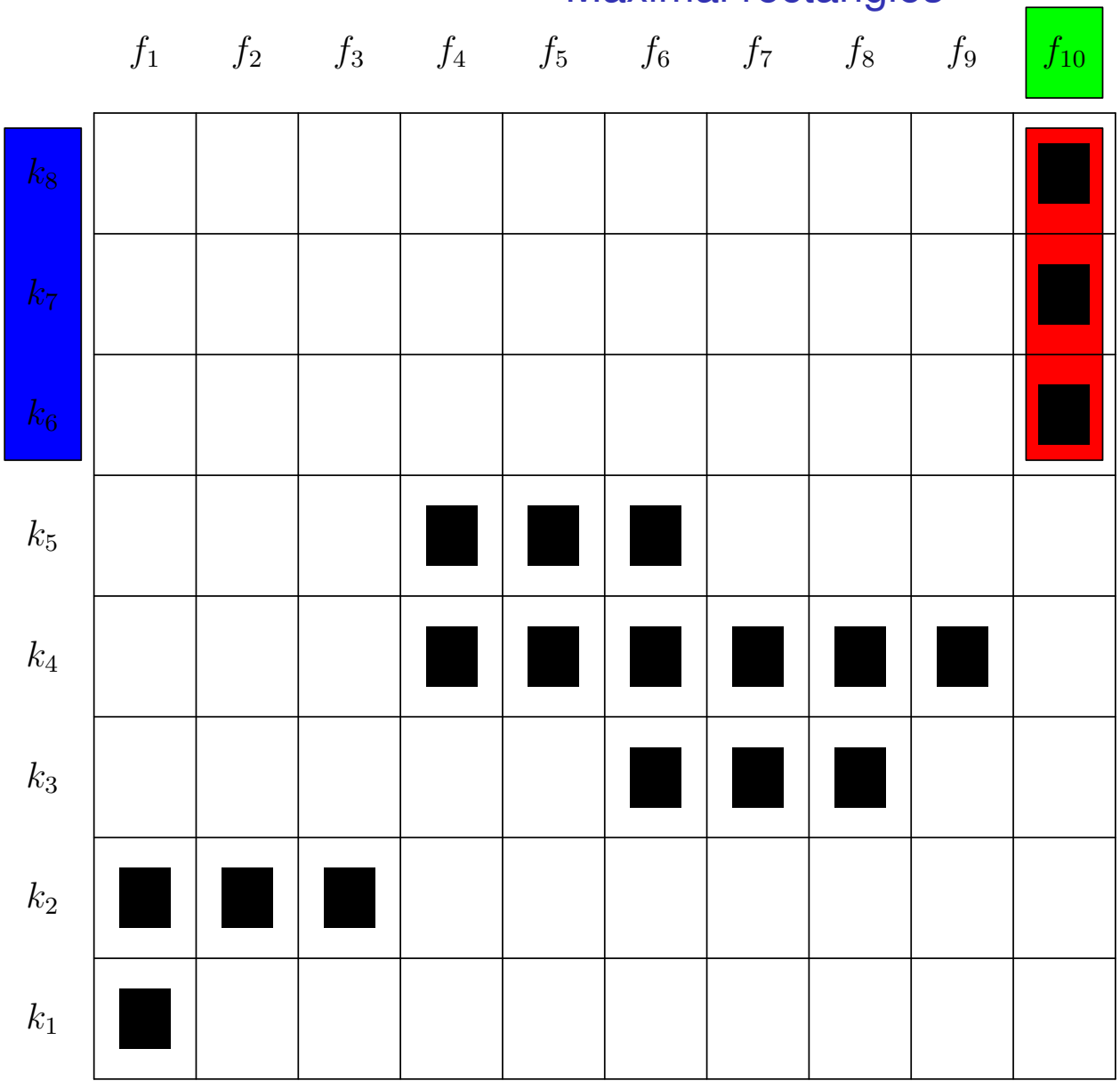
# Concepts

## Maximal rectangles



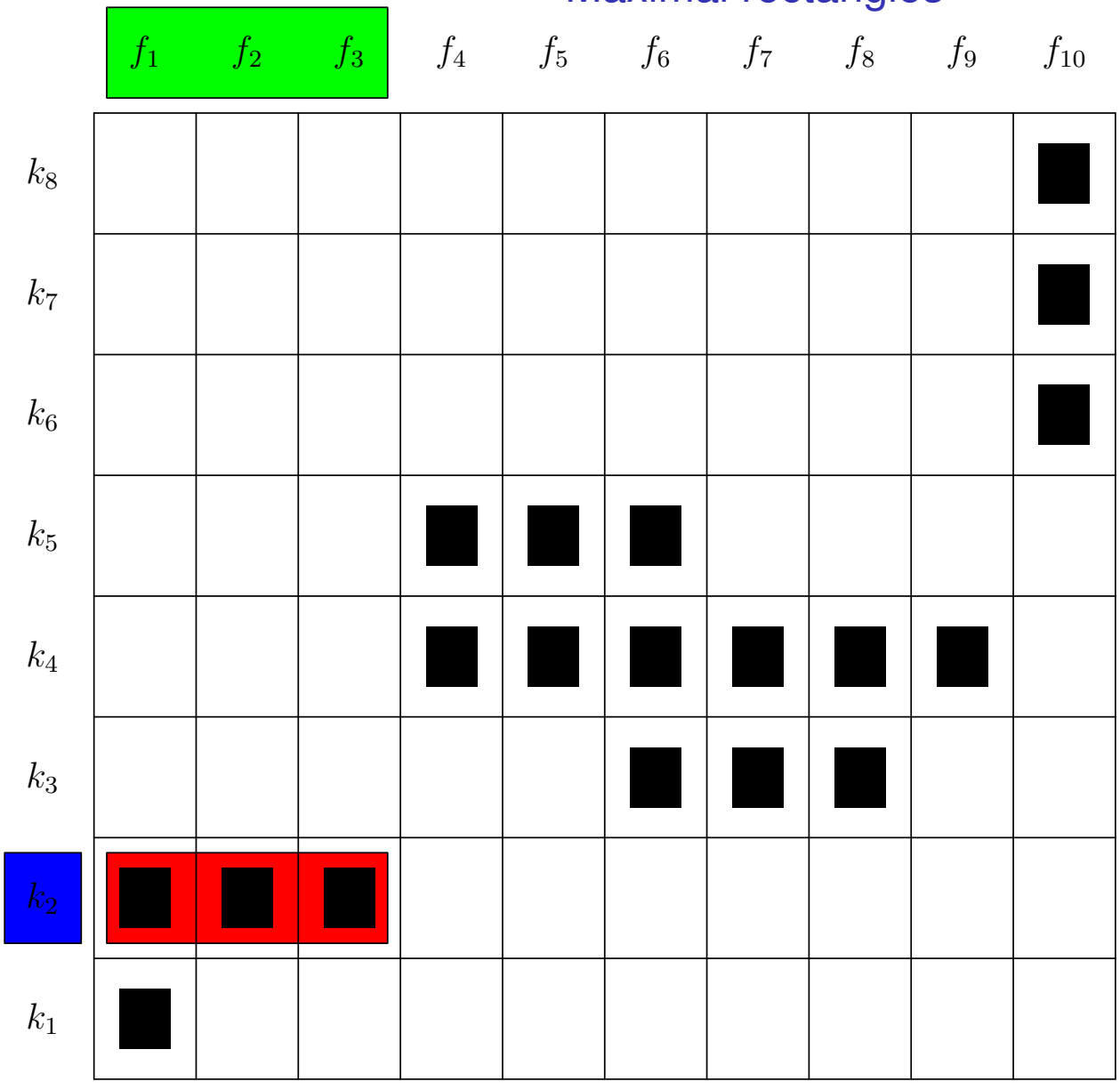
# Concepts

## Maximal rectangles



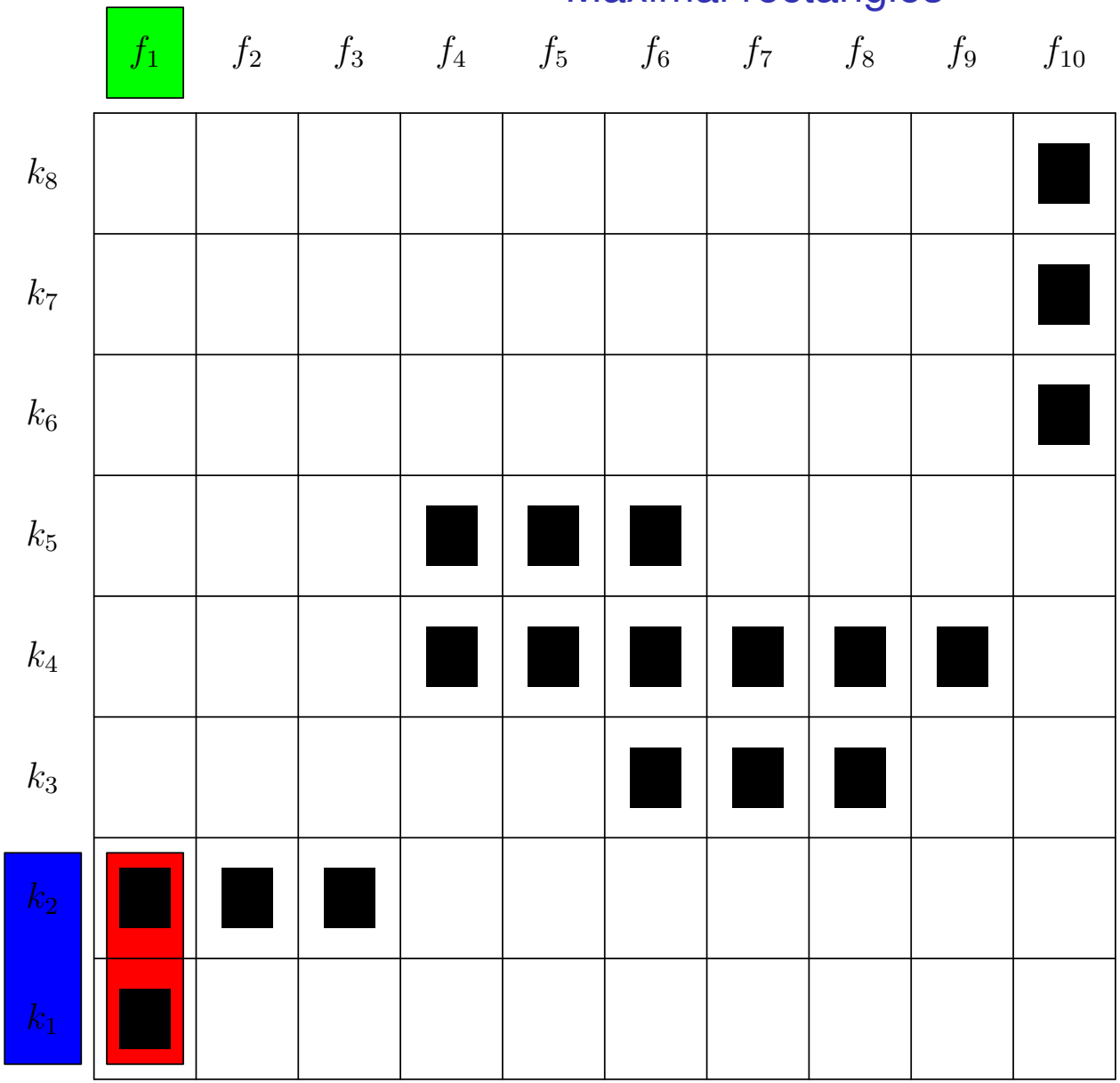
# Concepts

## Maximal rectangles

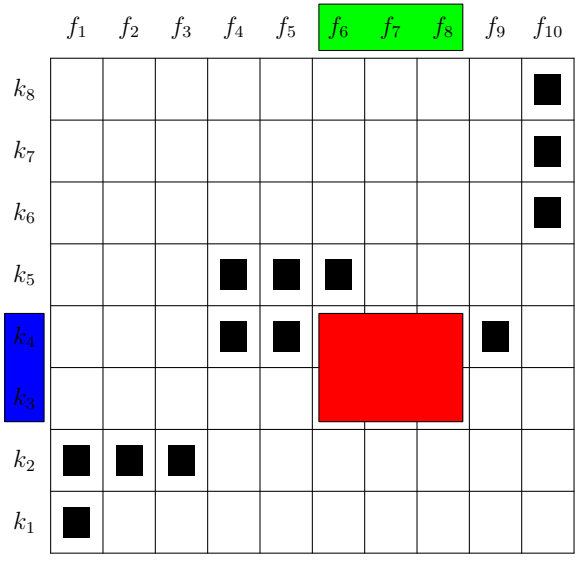


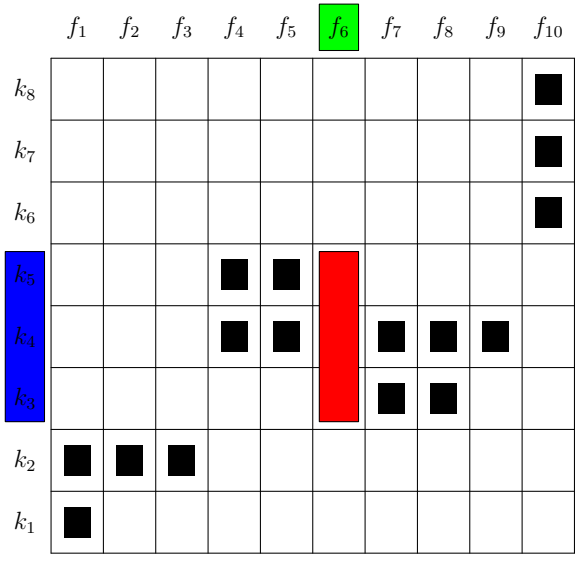
# Concepts

## Maximal rectangles

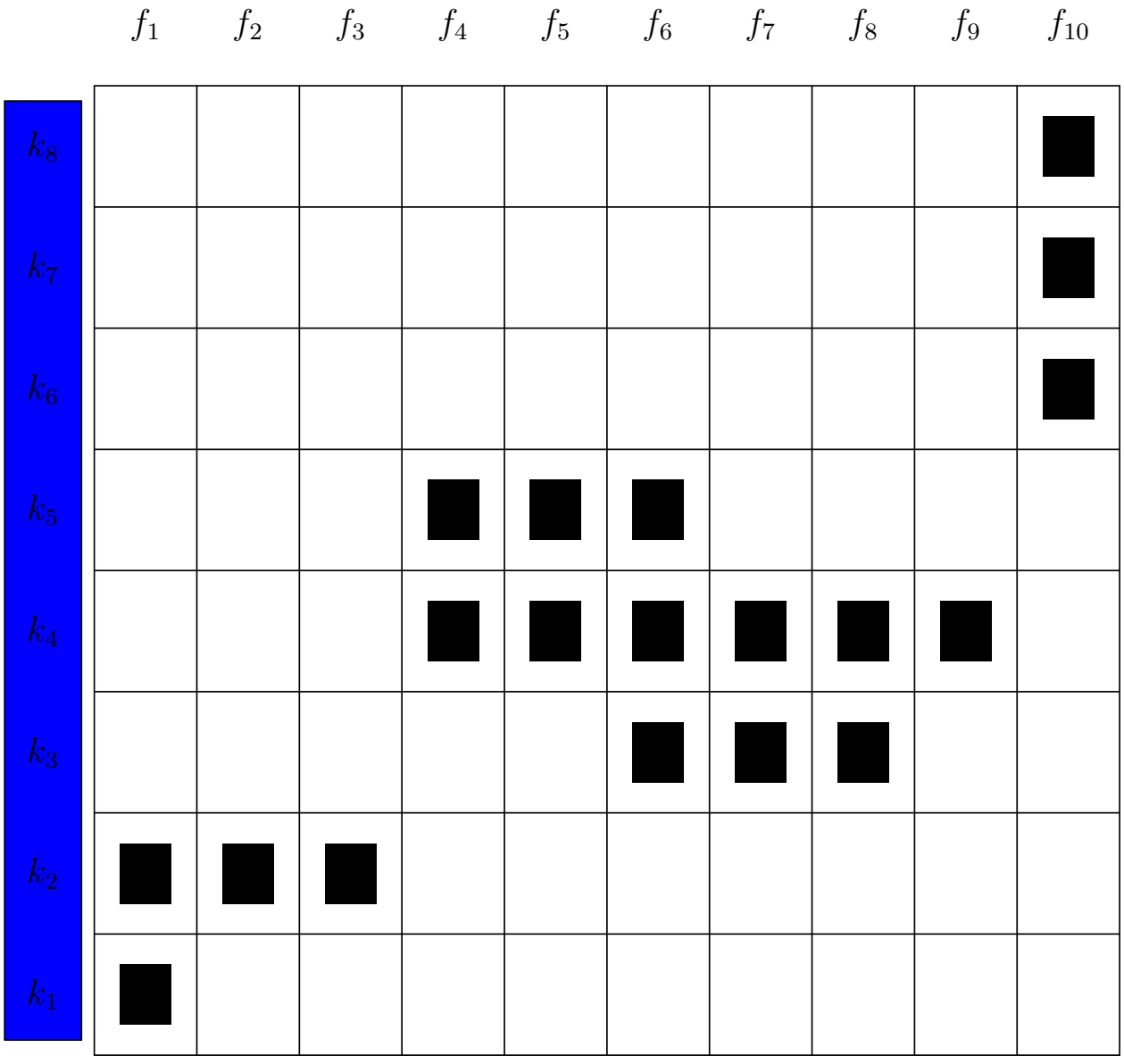


# Partial order

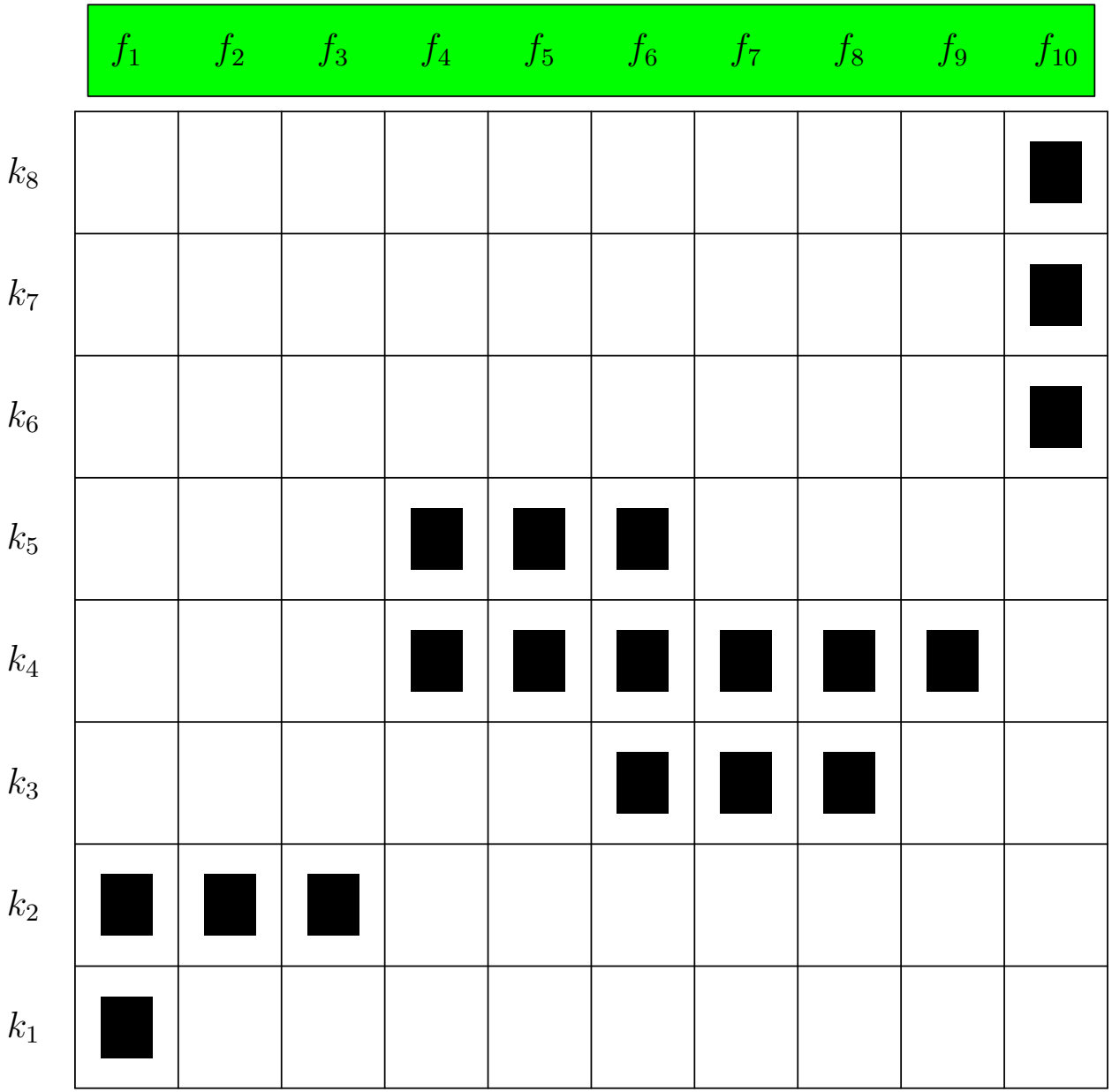


# Top and bottom

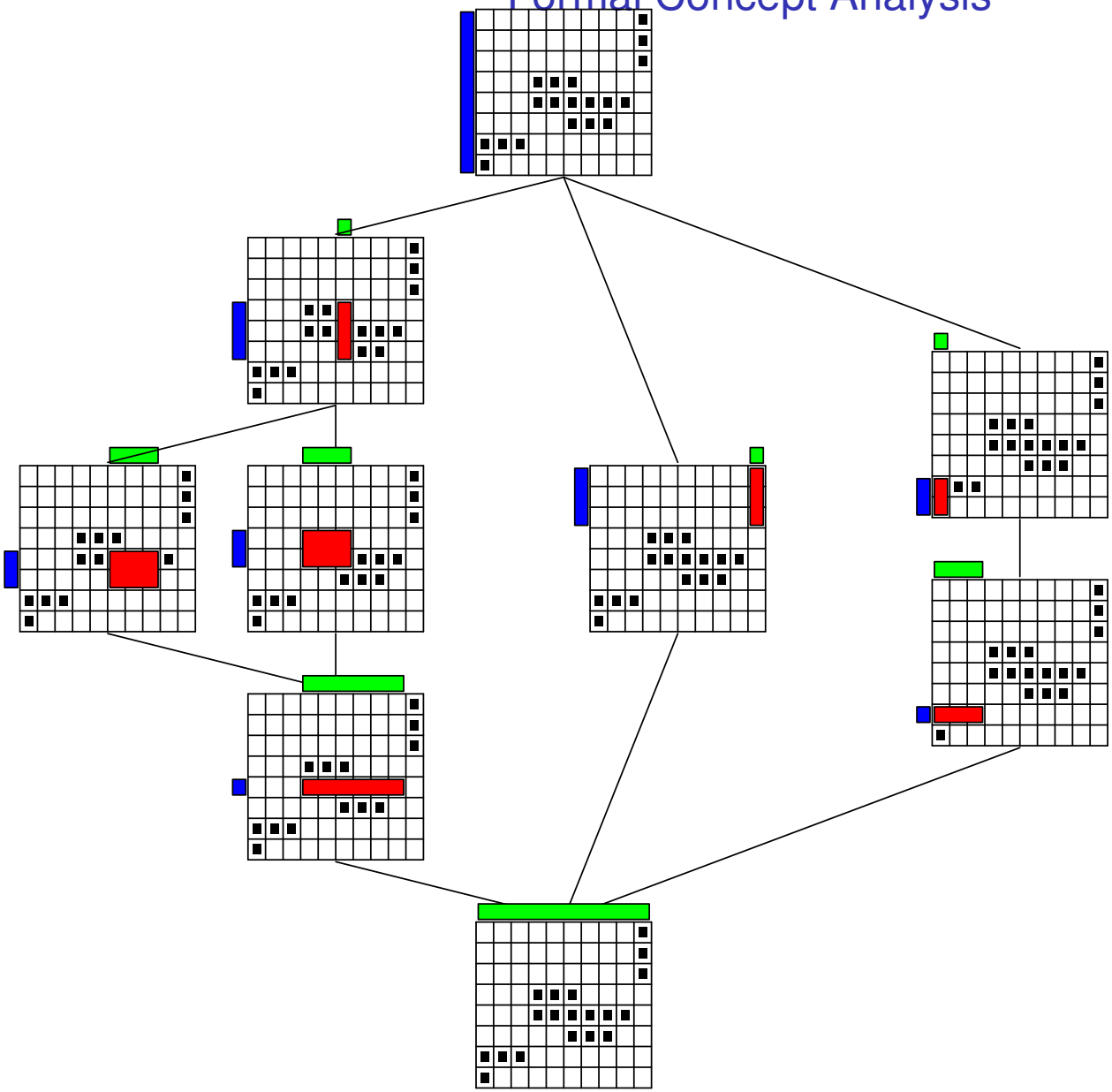


# Top and bottom



# Complete Lattice

## Formal Concept Analysis



# Rulon Wells

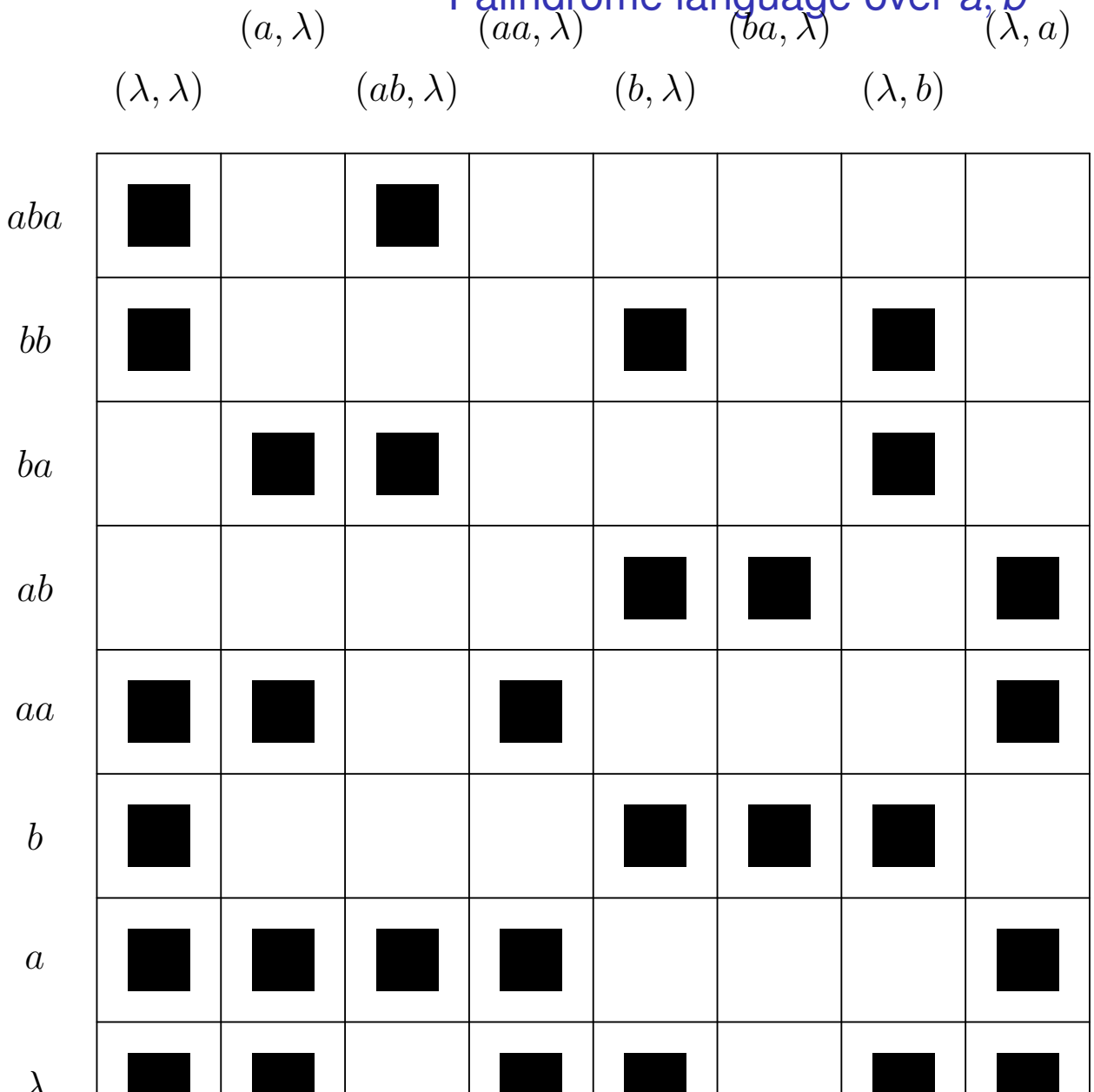
[Wells(1947)]

It is easy to define a focus-class embracing a large variety of sequence classes but characterized by only a few environments; it is also easy to define one characterized by a great many environments in which all its members occur but on the other hand poor in the number of diverse sequence-classes that it embraces. What is difficult, but far more important than either of the easy tasks, is to define focus-classes rich both in the number of environments chracterizing them and at the same time in the diversity of sequence classes that they embrace.

- Concepts high up in the lattice have a few contexts, but lots of strings
- Concepts low down have a larger number of contexts, but only a few strings.

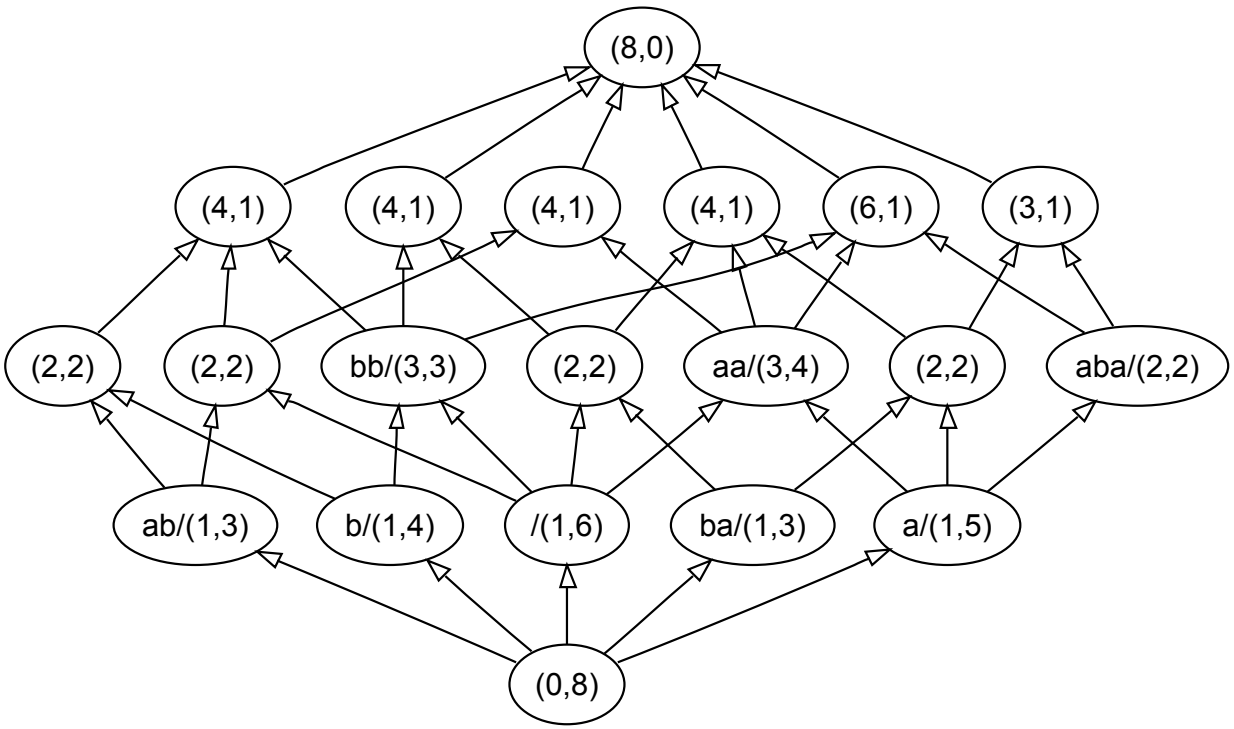
# Lattice

## Palindrome language over *a*, *b*



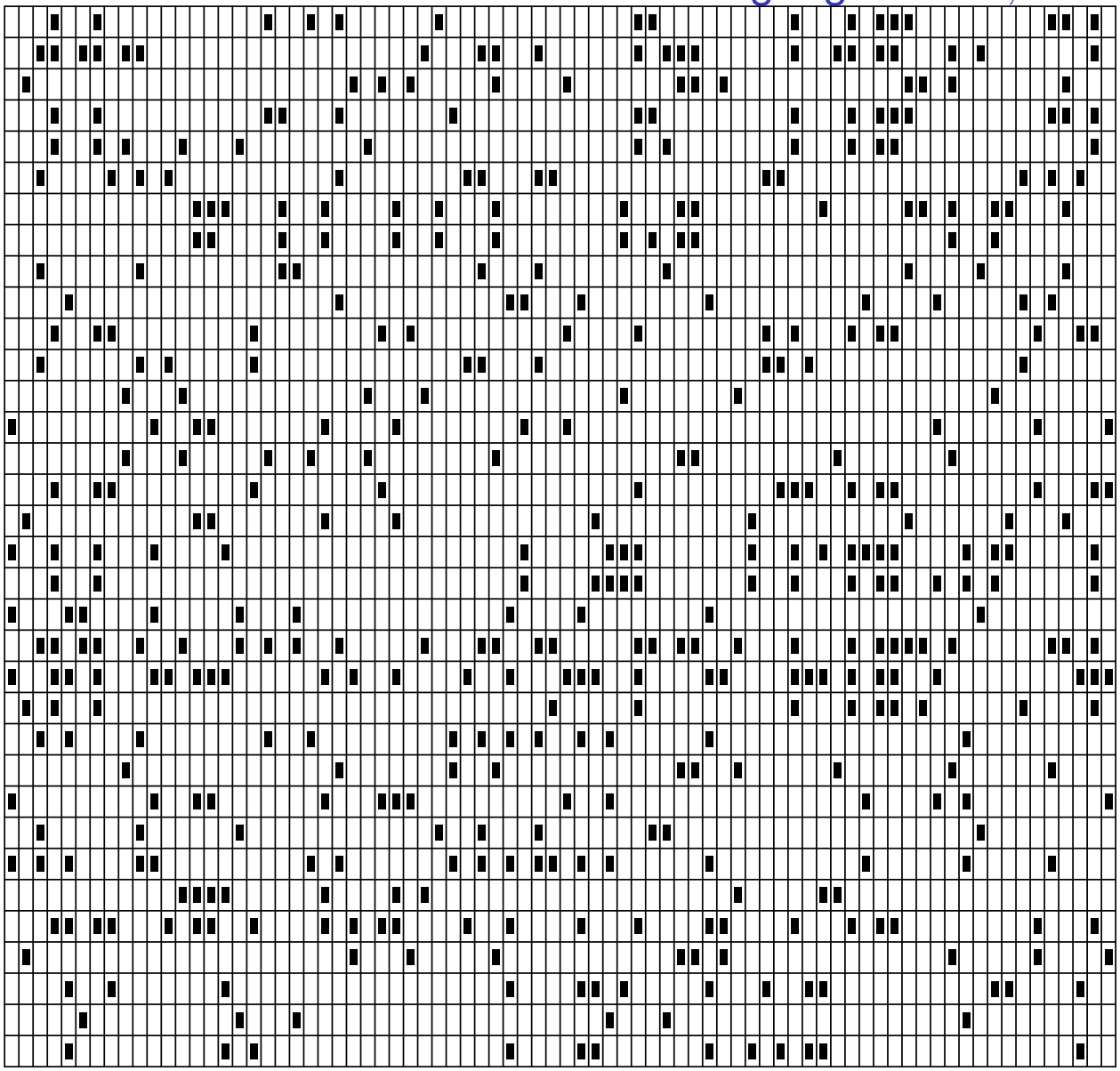
# Lattice

## Many rectangles

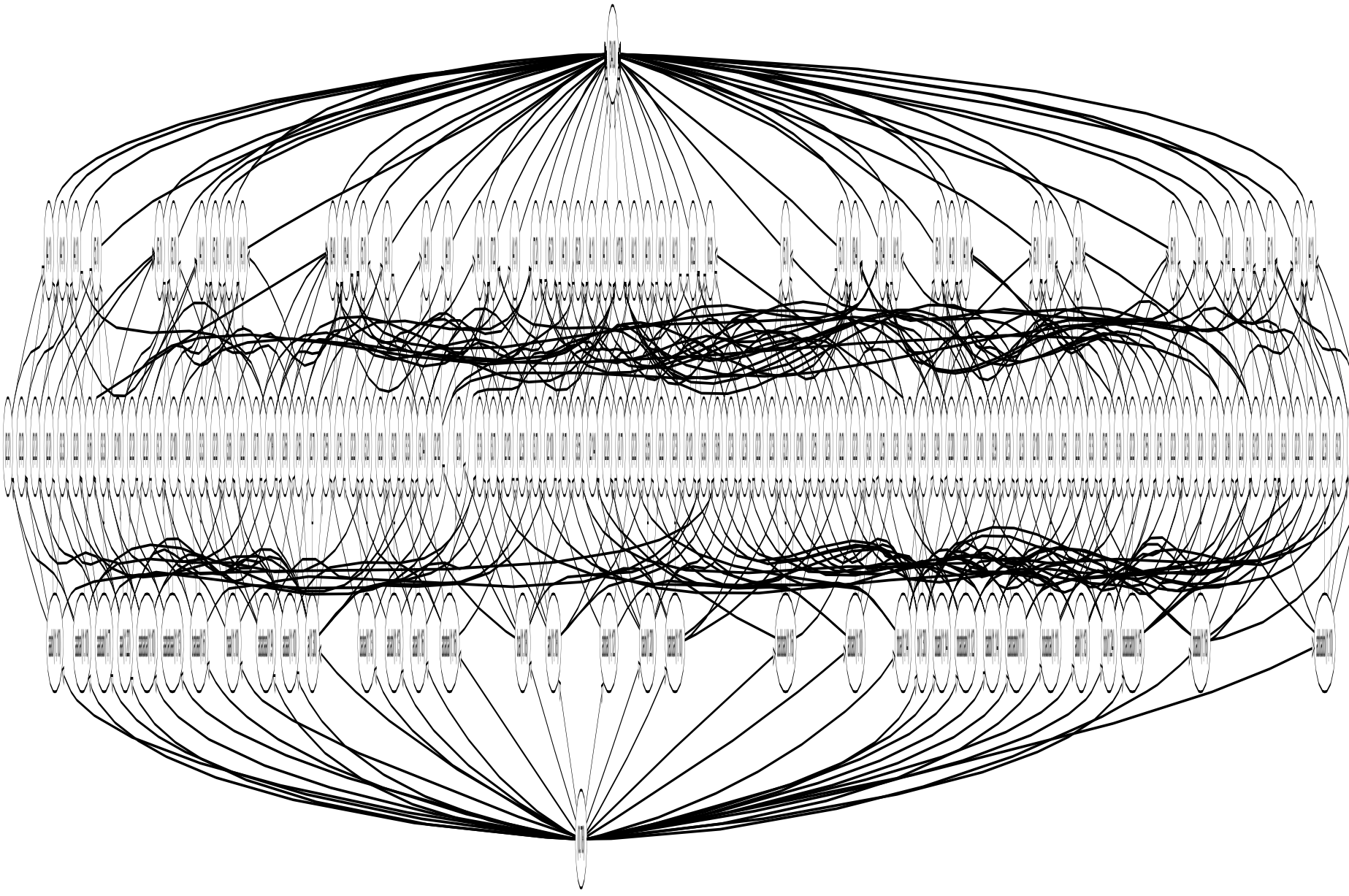


# Lattice

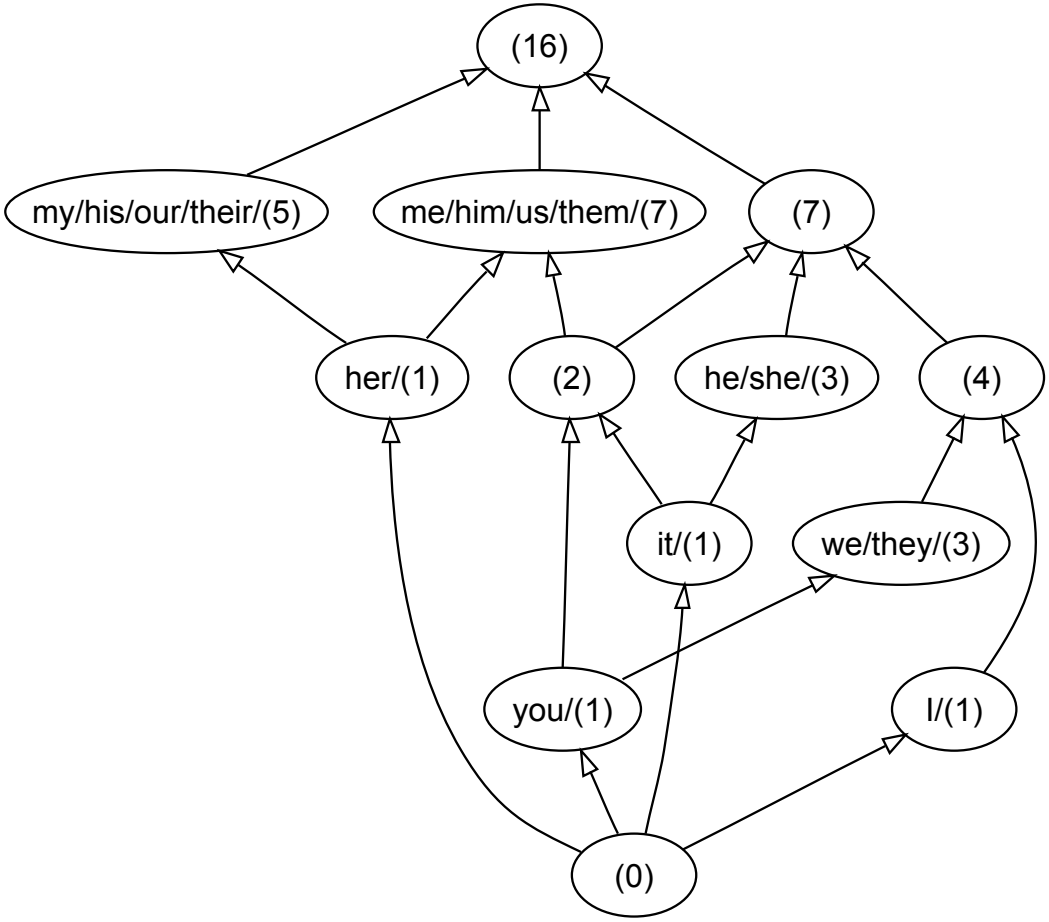
## Palindrome language over $a, b$



# Lattice



# Linguistic concepts



# Non-terminals

## On the left hand side

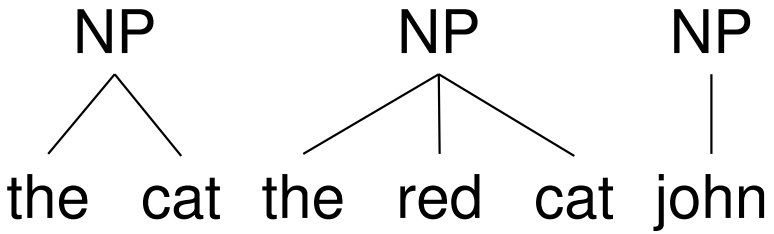
```
NP -> D N
NP -> D Adj N
NP -> N_proper
```

## On the right hand side

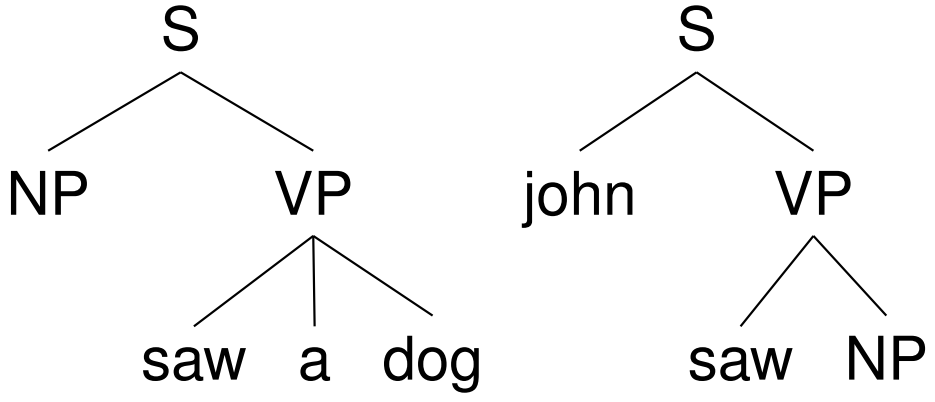
```
S -> NP VP
VP -> V_trans NP
```

# Examples

On the left hand side: substrings: $Y(NP)$

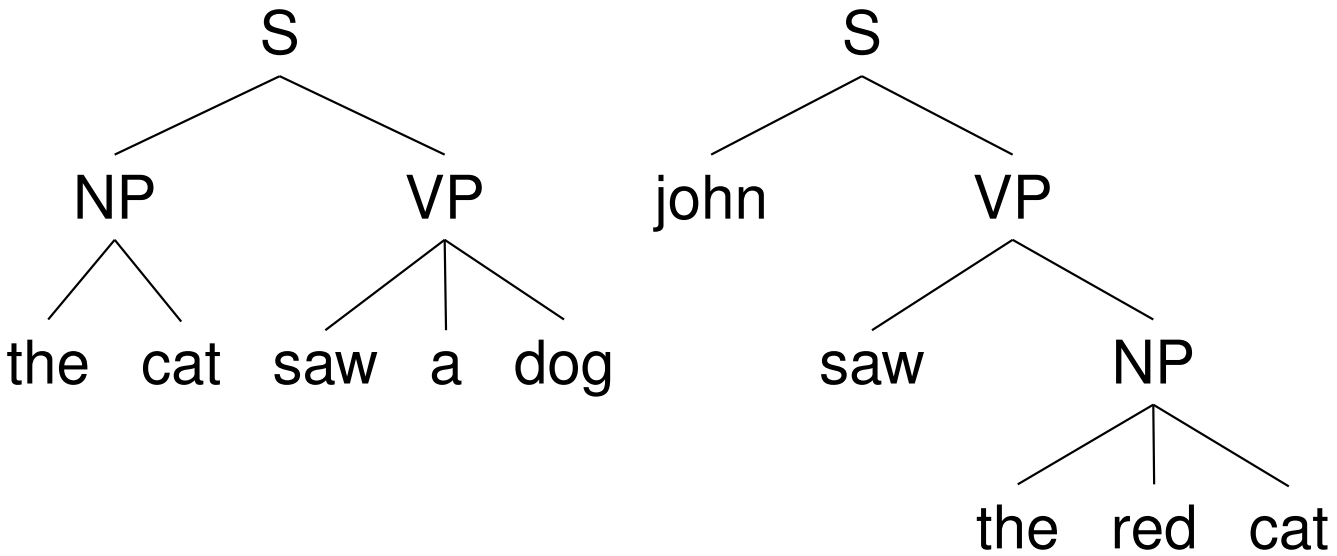



On the right hand side: contexts $C(NP)$



# Context free

The term “context-free” means we can combine any context with any substring. Non-terminals are rectangles.



# Minimal grammars

## Theorem: [Clark(2013a)]

The smallest grammar that generates a given language will have categories that are *maximal rectangles* in the context substring table.

# Context free grammar

## Representation

Non-terminals correspond to syntactic concepts.

# Context free grammar

## Representation

Non-terminals correspond to syntactic concepts.

## Rules

$[[X]] \rightarrow [[Y]][[Z]]$
If and only if $X^{\triangleleft} \supseteq Y^{\triangleleft} Z^{\triangleleft}$

# Bibliography I



R.C. Berwick, P. Pietroski, B. Yankama, and N. Chomsky.
Poverty of the stimulus revisited.
*Cognitive Science*, 35:1207–1242, 2011.


A. Clark.
The syntactic concept lattice: Another algebraic theory of the context-free languages?
*Journal of Logic and Computation*, 2013a.
doi: 10.1093/logcom/ext037.


Alexander Clark.
Learning trees from strings: A strong learning algorithm for some context free grammars.
*Journal of Machine Learning Research*, 2013b.
to appear.

# Bibliography II



D. Lewis.
Languages and language.
pages 3–35. Minneapolis, University of Minnesota Press, 1975.


Marvin C. Paull and Stephen H. Unger.
Structural equivalence of context-free grammars.
*Journal of Computer and System Sciences*, 2(4):427 – 463, 1968.
ISSN 0022-0000.
doi: 10.1016/S0022-0000(68)80037-6.
URL http://www.sciencedirect.com/science/article/pii/S0022000068800376.


R. S. Wells.
Immediate constituents.
*Language*, 23(2):81–117, 1947.

# Bibliography III


K. Wexler and P. W. Culicover.
*Formal Principles of Language Acquisition*.
MIT Press, Cambridge, MA, 1980.