Distributional Learning of Context-Free Grammars.

Alexander Clark

Department of Philosophy
King's College London
alexander.clark@kcl.ac.uk

14 November 2018
UCL

Outline

Outline

Machine learning

Standard machine learning problem

We learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ from a sequence of input-output pairs $\langle(\mathbf{x_1}, \mathbf{y_1}) \ldots (\mathbf{x_n}, \mathbf{y_n})\rangle$

Convergence

As $n \rightarrow \infty$ we want our hypothesis $\hat{f}$ to tend to f
Ideally we want $\hat{f} = f$.

Vector spaces

Standard two assumptions

1. Assume sets have some algebraic structure:
   - $\mathcal{X}$ is $\mathbb{R}^n$
   - $\mathcal{Y}$ is $\mathbb{R}$
2. Assume f satisfies some smoothness assumptions:
   - f is linear
   - or satisfies some Lipschitz condition: $|f(\mathbf{x_i}) - f(\mathbf{x_j}| \leq c|\mathbf{x_i} - \mathbf{x_j}|$

- The input examples are strings.
- No output (unsupervised learning!)
- Our representations are context-free grammars.

Context-Free Grammars

Context-Free Grammar

$G = \langle \Sigma, V, S, P \rangle$
$\mathcal{L}(G, A) = \{w \in \Sigma^* \mid A \stackrel{*}{\Rightarrow}_G w\}$

Example

$\Sigma = \{a, b\}, V = \{S\}$
$P = \{S \rightarrow ab, S \rightarrow aSb, S \rightarrow \epsilon\}$

$$\mathcal{L}(G, S) = \{a^n b^n \mid n \geq 0\}$$

Least fixed point semantics

[Ginsburg and Rice(1962)]

Interpret this as a set of equations in $\mathcal{P}(\Sigma^*)$

$$S = (a \circ b) \vee (a \circ S \circ b) \vee \epsilon$$

Least fixed point semantics

[Ginsburg and Rice(1962)]

Interpret this as a set of equations in $\mathcal{P}(\Sigma^*)$

$$S = (a \circ b) \vee (a \circ S \circ b) \vee \epsilon$$

- Ξ is the set of functions $V \to \mathcal{P}(\Sigma^*)$
- $\Phi_G : \Xi \to \Xi$

$$\Phi_G(\xi)[S] = (a \circ b) \vee (a \circ \xi(S) \circ b) \vee \epsilon$$

Least fixed point $\xi_G = \bigvee_n \Phi_G^n(\xi_\perp) = \{S \to \mathcal{L}(G, S)\}$

What Algebra?

Monoid: $\langle S, \circ, 1 \rangle$

$$\Sigma^*$$

What Algebra?

Monoid: $\langle S, \circ, 1 \rangle$

$$\Sigma^*$$

Complete Idempotent Semiring: $\langle S, \circ, 1, \vee, \perp \rangle$

$$\mathcal{P}(\Sigma^*)$$

Outline

Introduction

Weak Learning

Strong Learning

An Algebraic Theory of CFGs

Running example

Propositional logic

Alphabet

rain, snow, hot, cold, danger	$A_1, A_2, \ldots$
and, or, implies, iff	$\wedge, \vee, \rightarrow, \leftrightarrow$
not	$\neg$
open, close	$($, $)$

Running example

Propositional logic

Alphabet

rain, snow, hot, cold, danger	$A_1, A_2, \ldots$
and, or, implies, iff	$\wedge, \vee, \rightarrow, \leftrightarrow$
not	$\neg$
open, close	$(,)$

- rain
- open snow implies cold close
- open snow implies open not hot close close

Distributional Learning

[Harris(1964)]

- Look at the dog
- Look at the cat

Distributional Learning
[Harris(1964)]

- Look at the dog
- Look at the cat
- That cat is crazy

Distributional Learning
[Harris(1964)]

- Look at the dog
- Look at the cat
- That cat is crazy
- That dog is crazy

English counterexample

- I can swim
- I may swim
- I want a can of beer

English counterexample

- I can swim
- I may swim
- I want a can of beer
- *I want a may of beer

English counterexample

- She is Italian
- She is a philosopher
- She is an Italian philosopher

English counterexample

- She is Italian
- She is a philosopher
- She is an Italian philosopher
- *She is an a philosopher philosopher

Logic example

Propositional logic is *substitutable*:

- open rain and cold close
- open rain implies cold close
- open snow implies open not hot close

Logic example

Propositional logic is *substitutable*:

- open rain and cold close
- open rain implies cold close
- open snow implies open not hot close
- open snow and open not hot close

Formally

The Syntactic Congruence: a monoid congruence

Two nonempty strings u, v are congruent ($u \equiv_L v$) if for all $l, r \in \Sigma^*$

$$lur \in L \Leftrightarrow lvr \in L$$

We write $[u]$ for the congruence class of u.

Definition

L is substitutable if
$lur \in L, lvr \in L \Rightarrow u \equiv_L v$

Example

Input data $D \subseteq L$

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

One production for each example

- $S \rightarrow$ hot
- $S \rightarrow$ cold
- $S \rightarrow$ open hot or cold close
- $S \rightarrow$ open not hot close
- $S \rightarrow$ open hot and cold close
- $S \rightarrow$ open hot implies cold close
- $S \rightarrow$ open hot iff cold close
- $S \rightarrow$ danger
- $S \rightarrow$ rain
- $S \rightarrow$ snow

A trivial grammar

Input data D

$D = \{w_1, w_2, \ldots, w_n\}$ are nonempty strings.

Starting grammar

$S \to w_1, S \to w_2, \ldots, S \to w_n$
$\mathcal{L}(G) = D$

A trivial grammar

Input data D

$D = \{w_1, w_2, \ldots, w_n\}$ are nonempty strings.

Starting grammar

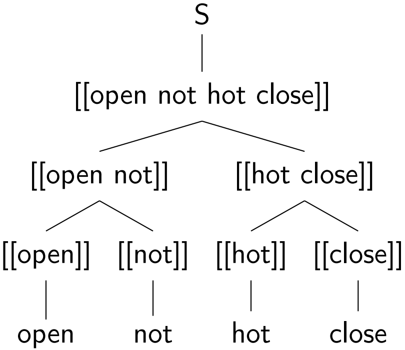
$S \to w_1, S \to w_2, \ldots, S \to w_n$
$\mathcal{L}(G) = D$

Binarise this every way

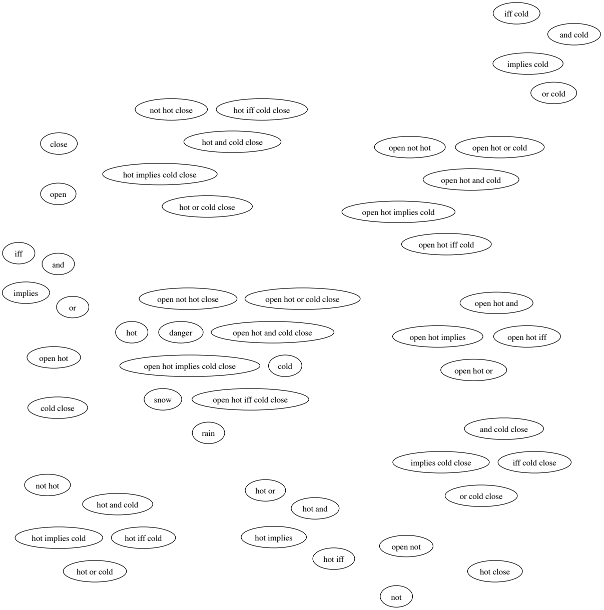
One nonterminal $[[w]]$ for every substring w.

- $[[a]] \to a$
- $S \to \{w\}$, $w \in D$
- $[[w]] \to [[u]][[v]]$ when $w = u \cdot v$

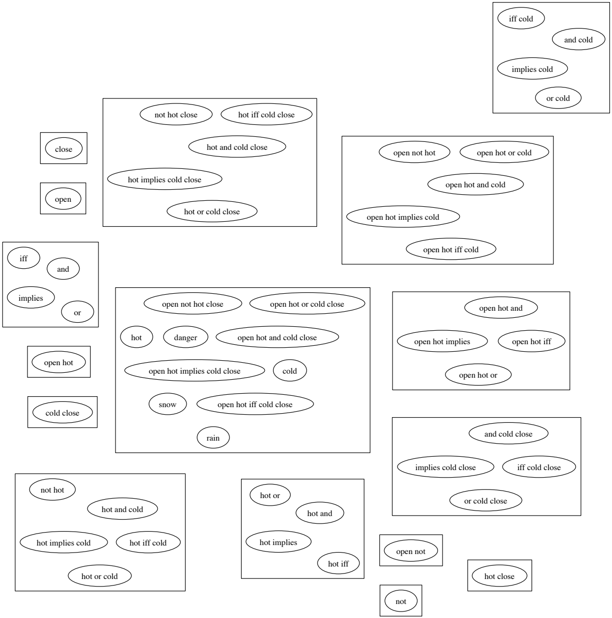
$$\mathcal{L}(G, [[w]]) = \{w\}$$


S
[[open not hot close]]
[[open not]]
[[hot close]]
[[open]]
[[not]]
[[hot]]
[[close]]
open
not
hot
close

Nonterminal for each substring



Nonterminal for each cluster



Productions

Observation

If $w = u \cdot v$ then $[w] \supseteq [u] \cdot [v]$

Productions

Observation

If $w = u \cdot v$ then $[w] \supseteq [u] \cdot [v]$

Add production

$[[w]] \rightarrow [[u]][[v]]$

Productions

Observation

If $w = u \cdot v$ then $[w] \supseteq [u] \cdot [v]$

Add production

$[[w]] \rightarrow [[u]][[v]]$

Consequence

If L is substitutable, then

$$\mathcal{L}(G, [[w]]) \subseteq [w]$$

$$\mathcal{L}(G) \subseteq L$$

Theorem [Clark and Eyraud(2007)]

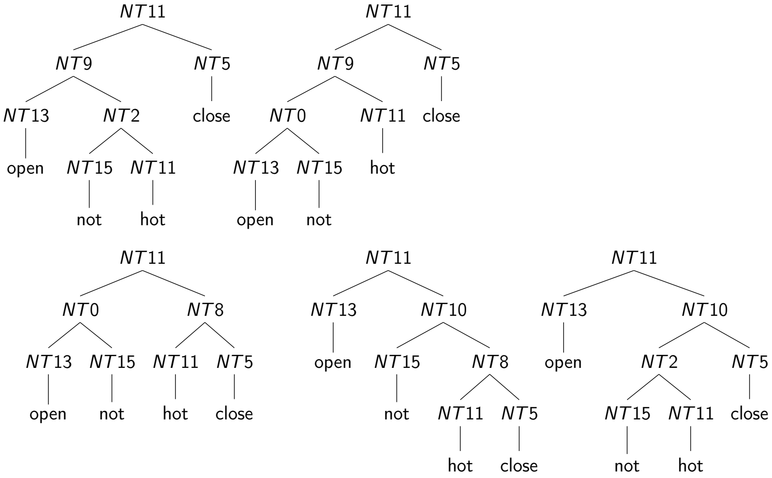
- If the language is a substitutable context-free language, then the hypothesis grammar will converge to a correct grammar.
- Efficient; provably correct

Theorem [Clark and Eyraud(2007)]

- If the language is a substitutable context-free language, then the hypothesis grammar will converge to a correct grammar.
- Efficient; provably correct

But the grammar may be different for each input data set!

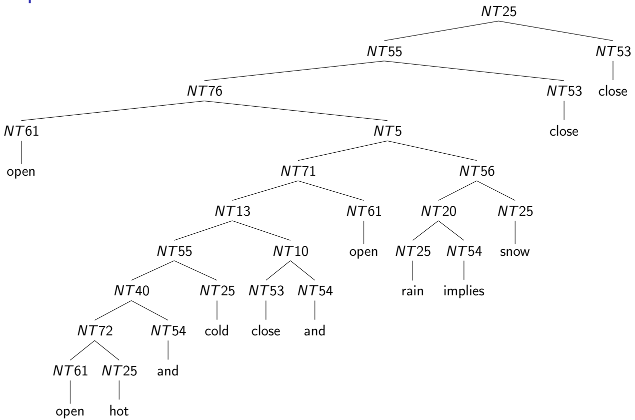
open not hot close



Larger data set: 92 nonterminals, 435 Productions

open open hot and cold close and open rain implies snow close close

327204 parses



Outline

Strong Learning

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

Strong Learning

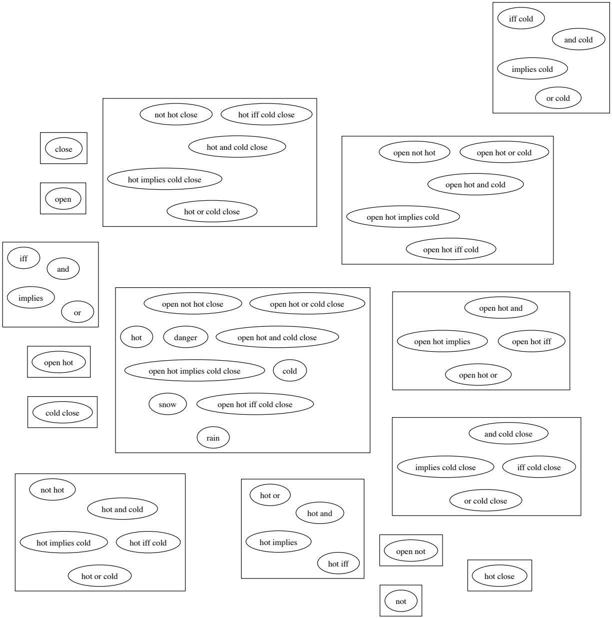
Target class of grammars

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

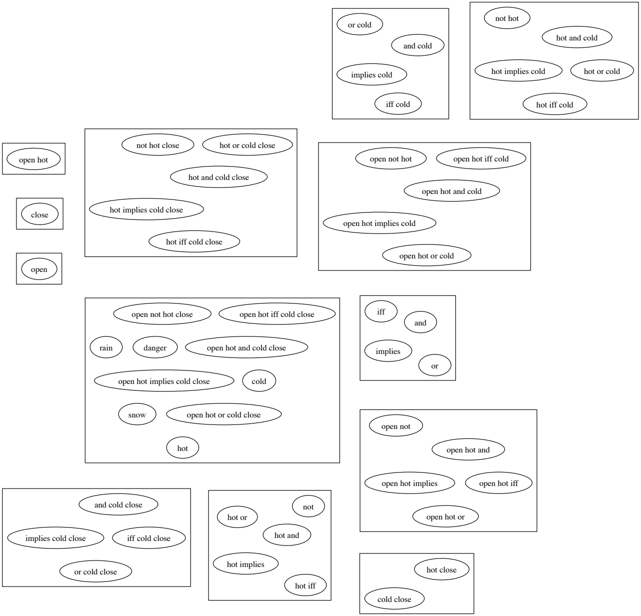
Strong learning

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 $\hat{G} \equiv G_*$

Inaccurate clusters



Correct congruence classes



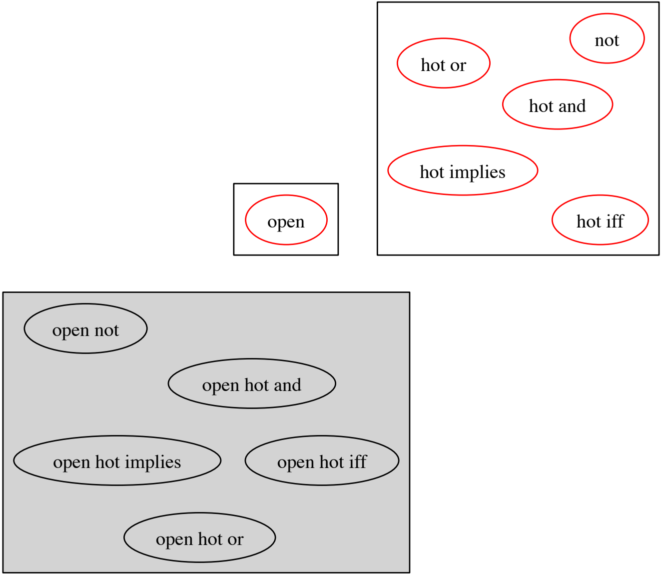
Myhill-Nerode Theorem

A language has a finite number of congruence classes if and only if it is regular.

Myhill-Nerode Theorem

A language has a finite number of congruence classes if and only if it is regular.

We need some principled way of picking a finite collection of “good” congruence classes.


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

Definition

Definition

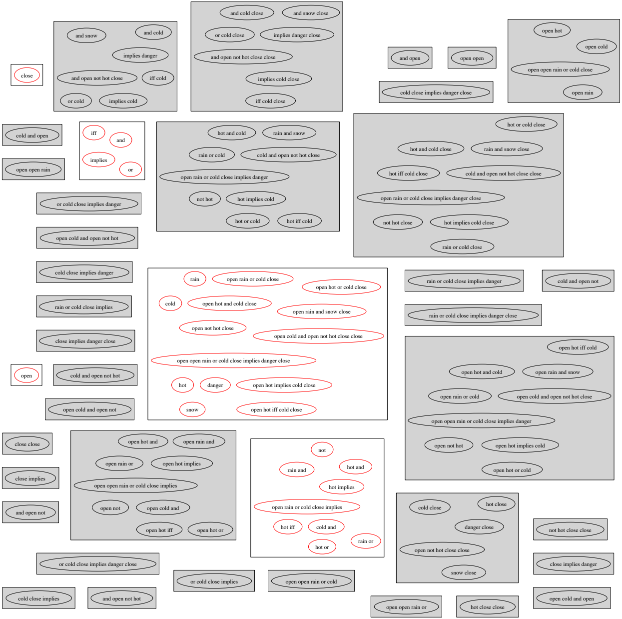
A congruence class X is composite if there are two congruence classes Y, Z such that $X = YZ$.
(and neither Y nor Z is the class $[\lambda]$)

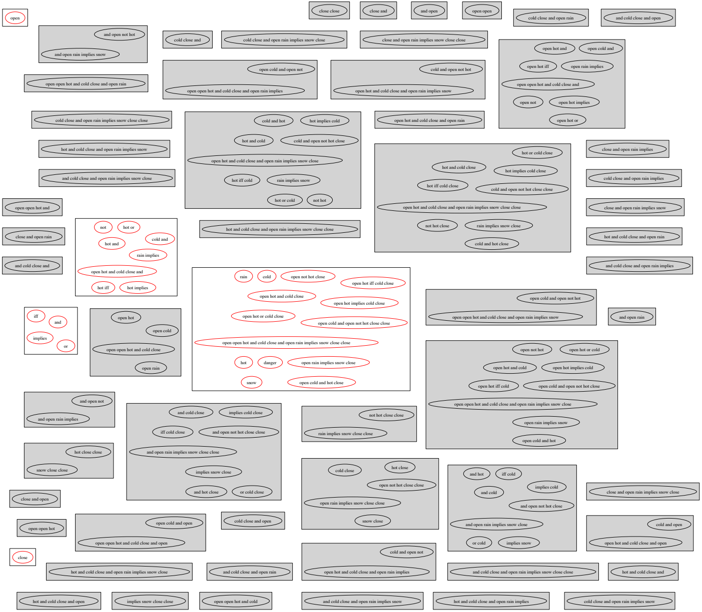
Definition

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

The whole is greater than the sum of the parts


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


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

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 ...

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 $Q = P_1 \ldots P_n$

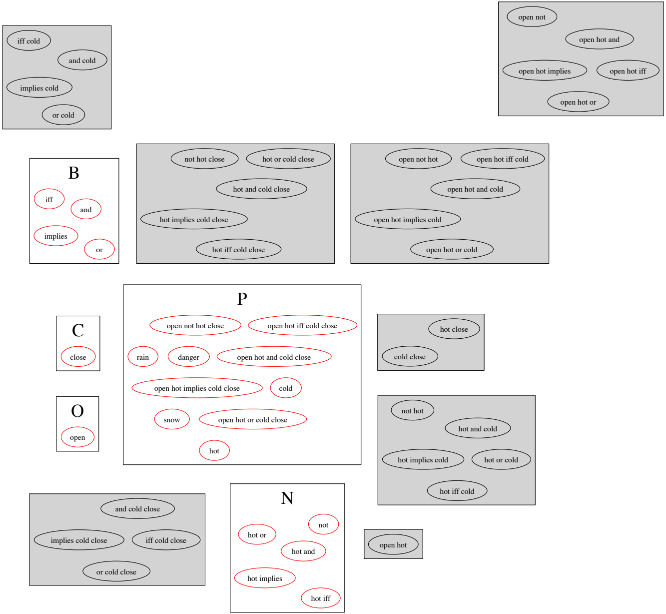
Intuition

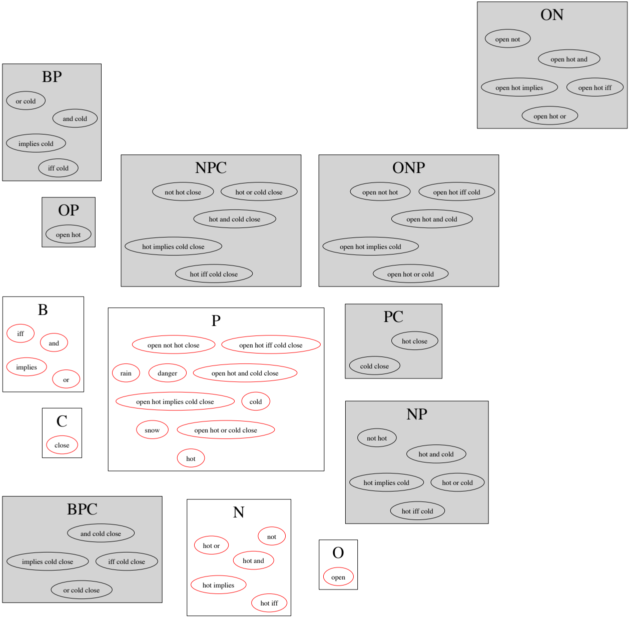
If $X = YZ$, and we have a rule

$$P \to QXR$$

then we can change it to

$$P \to QYZR$$


iff cold
and cold
implies cold
or cold
open not
open hot and
open hot implies
open hot iff
open hot or
B
iff
and
implies
or
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
P
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
C
close
O
open
hot close
cold close
not hot
hot and cold
hot implies cold
hot or cold
hot iff cold
and cold close
implies cold close
iff cold close
or cold close
N
not
hot or
hot and
hot implies
hot iff
open hot



ON
open not
open hot and
open hot implies
open hot iff
open hot or
BP
or cold
and cold
implies cold
iff cold
NPC
not hot close
hot or cold close
hot and cold close
hot implies cold close
hot iff cold close
ONP
open not hot
open hot iff cold
open hot and cold
open hot implies cold
open hot or cold
OP
open hot
B
iff
and
implies
or
P
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
PC
hot close
cold close
C
close
NP
not hot
hot and cold
hot implies cold
hot or cold
hot iff cold
BPC
and cold close
implies cold close
iff cold close
or cold close
N
not
hot or
hot and
hot implies
hot iff
O
open

Productions

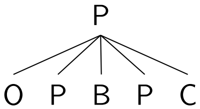
We need non-binary rules.

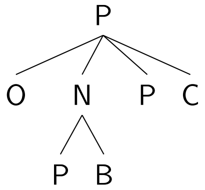
Correct productions

$P_0 \rightarrow P_1 \ldots P_k$
where $P_0 \supsetneq P_1 \ldots P_k$

Infinite number of correct productions

- $N \rightarrow PB$
- $P \rightarrow ONPC$
- $P \rightarrow OPBPC$
- $P \rightarrow ONONPCC$
- $\ldots$


P
O P B P C


P
O
N
P
C
P
B

Productions

Valid productions

- Correct productions where the right hand side does not contain the right hand side of a valid production.
- If there are n primes then there are at most n^2 valid productions.

Examples

- $N \rightarrow PB$
- $P \rightarrow ONPC$

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.

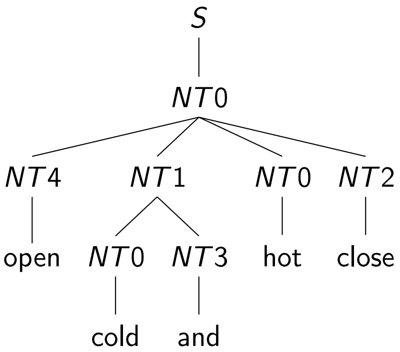
Theorem [Clark(2014)]

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

- From positive examples
- Identification in the limit
- Strongly: converges structurally
- Using polynomial time and data

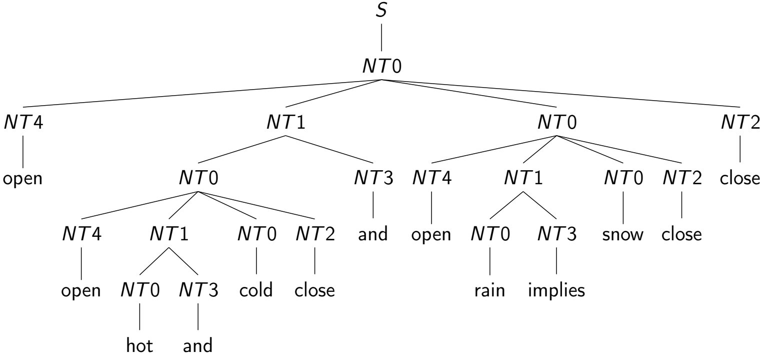
Running example

(verbatim output from implementation)



open open hot and cold close and open rain implies snow close close

1 parse



Outline

Contexts

A context is a string with a hole:

$$l\Box r$$

Derivation contexts

The derivation contexts of CFGs are just string contexts:

$$S \overset{*}{\Rightarrow}_G lNr$$

Definition

Filling the hole

$l\Box r \odot u = lur$

A factorisation of a language L

C is a set of contexts; S is a set of strings

$$C \odot S \subseteq L$$

Context free grammars

Contexts and yields

$\mathcal{L}(G, N) = \{w \in \Sigma^* \mid N \overset{*}{\Rightarrow} w\}$
$\mathcal{C}(G, N) = \{l \square r \mid S \overset{*}{\Rightarrow} lNr\}$.

Nonterminals in a context-free grammar

$$\mathcal{C}(G, N) \odot \mathcal{L}(G, N) \subseteq L$$

Context free grammars

Contexts and yields

$\mathcal{L}(G, N) = \{w \in \Sigma^* \mid N \stackrel{*}{\Rightarrow} w\}$
$\mathcal{C}(G, N) = \{l\Box r \mid S \stackrel{*}{\Rightarrow} lNr\}$.

Nonterminals in a context-free grammar

$$\mathcal{C}(G, N) \odot \mathcal{L}(G, N) \subseteq L$$

We can reverse this process and go from a collection of decompositions back to a CFG.

Polar maps

If S is a set of strings:

$$S^{\triangleright} = \{l\Box r \mid \forall u \in S, lur \in L\} \tag{1}$$

If C is a set of contexts:

$$C^{\triangleleft} = \{u \in \Sigma^* \mid \forall l\Box r \in C, lur \in L\} \tag{2}$$

Polar maps

If S is a set of strings:

$$S^{\triangleright} = \{l\Box r \mid \forall u \in S, lur \in L\} \tag{1}$$

If C is a set of contexts:

$$C^{\triangleleft} = \{u \in \Sigma^* \mid \forall l\Box r \in C, lur \in L\} \tag{2}$$

$\cdot^{\triangleright}$ and $\cdot^{\triangleleft}$ form a Galois connection between sets of strings and sets of contexts.

Closed sets of strings

- $\cdot^{\triangleright\triangleleft}$ is a closure operator on the sets of strings;
- $X^{\triangleright} = Y^{\triangleright}$ is a CIS-congruence;
- L is always closed.

Closed sets of strings

- $\cdot^{\triangleright\triangleleft}$ is a closure operator on the sets of strings;
- $X^{\triangleright} = Y^{\triangleright}$ is a CIS-congruence;
- L is always closed.

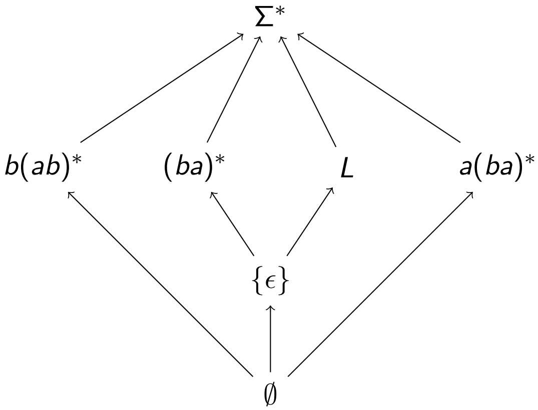
The syntactic concept lattice

The set of all closed sets of strings form a complete idempotent semiring: $\mathfrak{B}(L)$.

(A generalisation of the syntactic monoid; the collection of maximal decompositions into strings and contexts.)

L is regular iff $\mathfrak{B}(L)$ is finite

$L = (ab)^*$



Recognising a language

Definition

We say that a CIS B recognizes L if there is a surjective morphism h from $\mathcal{P}(\Sigma^*) \to B$ such that $h^*(h(L)) = L$, where h^* is the residual of h.

Recognising a language

Definition

We say that a CIS B recognizes L if there is a surjective morphism h from $\mathcal{P}(\Sigma^*) \to B$ such that $h^*(h(L)) = L$, where h^* is the residual of h.

Given a CIS B and a homomorphism $h : \mathcal{P}(\Sigma^*) \to B$, we can define a new grammar $\phi_h(G)$ by merging nonterminals M, N if

$$h(\mathcal{L}(G, M)) = h(\mathcal{L}(G, N))$$

Theorem

Let G be a CFG over Σ and h a homomorphism $h : \mathcal{P}(\Sigma^*) \to B$. Then

- $\phi_h(G)$ defines the same language as G iff
- B recognizes L through h

Theorem

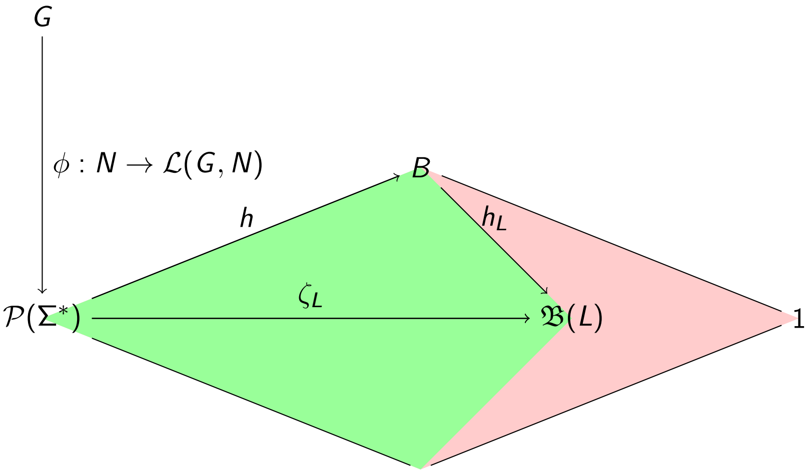
Let G be a CFG over Σ and h a homomorphism $h : \mathcal{P}(\Sigma^*) \to B$.
Then

- $\phi_h(G)$ defines the same language as G iff
- B recognizes L through h

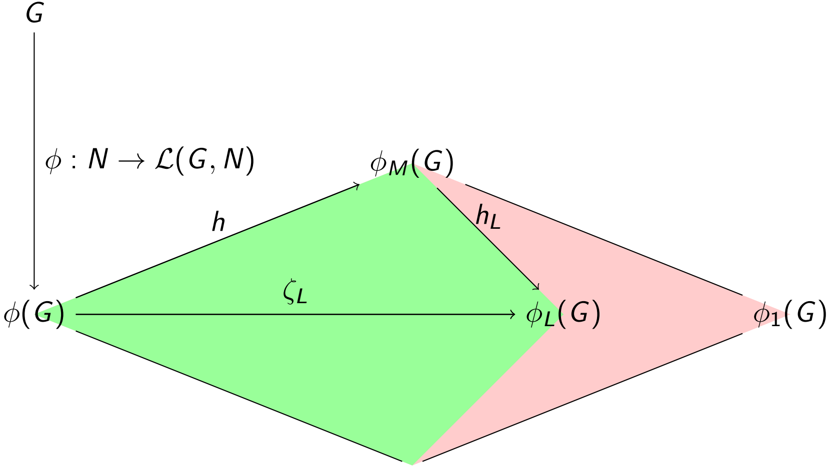
Uniqueness

There is a unique 'smallest' CIS that recognizes L: which is $\mathfrak{B}(L)$.

The universal morphism



The universal cfg-morphism



[Clark(2013)]

Mergeable nonterminals

If

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

then we can merge M and N without increasing the language defined by G,

[Clark(2013)]

Mergeable nonterminals

If

$$\mathcal{L}(G, M)^{\rhd\lhd} = \mathcal{L}(G, N)^{\rhd\lhd}$$

then we can merge M and N without increasing the language defined by G,

Minimal grammars correspond to maximal factorisations

A grammar without mergable nonterminals will have nonterminals that correspond to closed sets of strings.

Conclusion

- We can learn context-free grammars weakly decomposing strings into contexts and substrings.
- Minimal grammars will correspond to maximal decompositions.
- We can learn grammars strongly by identifying structure in some canonical algebras associated with the languages:
  - the syntactic monoid
  - the syntactic concept lattice.
- The same approach applies to Multiple Context-Free Grammars, a mildly context-sensitive grammar formalism.

Bibliography

Alexander Clark.
The syntactic concept lattice: Another algebraic theory of the context-free languages?
Journal of Logic and Computation, 2013.
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, 14:3537–3559, 2014.
URL http://jmlr.org/papers/v14/clark13a.html.

Alexander Clark and Rémi Eyraud.
Polynomial identification in the limit of substitutable context-free languages.
Journal of Machine Learning Research, 8:1725–1745, August 2007.

S. Ginsburg and H.G. Rice.
Two families of languages related to ALGOL.
Journal of the ACM (JACM), 9(3):350–371, 1962.

Zellig Harris.
Distributional structure.
In J. A. Fodor and J. J. Katz, editors, *The structure of language: Readings in the philosophy of language*, pages 33–49.
Prentice-Hall, 1964.