An(other) algebraic theory of the context-free languages?

Alexander Clark

Department of Philosophy
King's College London
alexsclark@gmail.com

July 2013
FSMNLP St Andrews

Outline

Outline

FSMNLP

1. Regular languages are awesome.
2. Natural languages aren't regular.

Regular languages

Classical Automata Theory

- Equivalence of regular expressions, DFAs and NFAs
- Myhill-Nerode theorem
- Canonical representations: minimal DFA

Algorithms

- Determinisation and Minimisation algorithms
- FST and weighted FST libraries
- Angluin's learnability results

Overview

Current NLP practice uses two ingredients:

1. A 50-year old representation:

   Finite Automata McCulloch and Pitts (1943)
   Categorial Grammars Bar-Hillel (1953)
   Context Free Grammars Chomsky (1956)
   Type-logical Grammars Lambek (1958)
   Dependency grammars Tesnière (1959)

2. Modern machine learning techniques:
   - Max-margin approaches and SVNs
   - Non-parametric Bayesian plus Markov Chain Monte Carlo
   - Deep belief networks
   - Spectral learning algorithms
   - ...

Goals

Questions

- Can we redo the theory of context-free languages to make it as useful as the theory of regular languages?
- And can we do the same later for the mildly context-sensitive language hierarchy?
- Can we use this to design algorithms for learning, minimizing, determinising, testing equivalence, . . . of context free grammars?

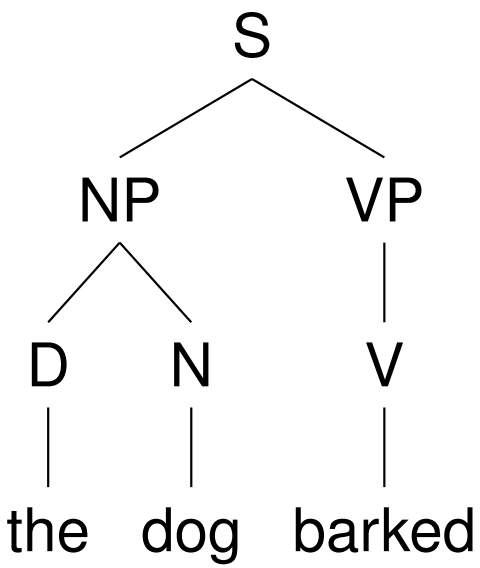
Goals

Questions

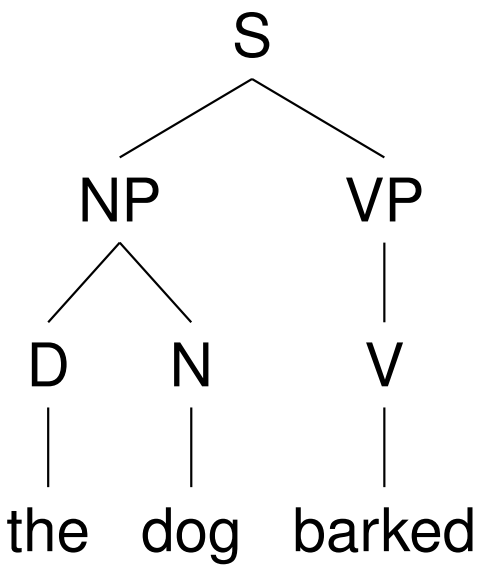
- Can we redo the theory of context-free languages to make it as useful as the theory of regular languages?
- And can we do the same later for the mildly context-sensitive language hierarchy?
- Can we use this to design algorithms for learning, minimizing, determinising, testing equivalence, . . . of context free grammars?

Maybe.

A naive question in linguistics



A naive question in linguistics



Question

What does this mean? What do the symbols mean?

Definitions

Context-free grammar

A tuple $\langle \Sigma, V, S, P \rangle$, where

- Σ is a nonempty finite set of terminal symbols
- V is a finite set of nonterminal symbols disjoint from Σ
- $S \in V$ is the start symbol
- $P \subset V \times (V \cup \Sigma)^+$ is a set of productions written $N \to \alpha$

Definitions

Context-free grammar

A tuple $\langle \Sigma, V, S, P \rangle$, where

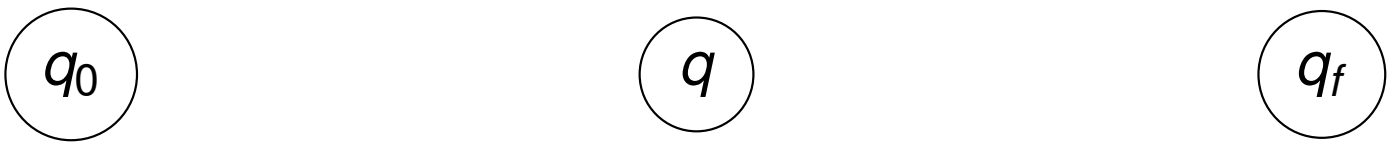
- Σ is a nonempty finite set of terminal symbols
- V is a finite set of nonterminal symbols disjoint from Σ
- $S \in V$ is the start symbol
- $P \subset V \times (V \cup \Sigma)^+$ is a set of productions written $N \to \alpha$

Questions

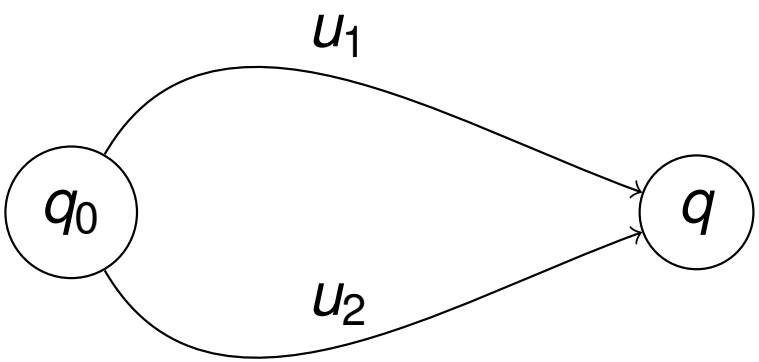
What are the elements of Σ? What are the elements of $V = \{N, P, Q \dots\}$? What do they mean? What does S mean? What does a production like $N \to aPQ$ mean?

Outline

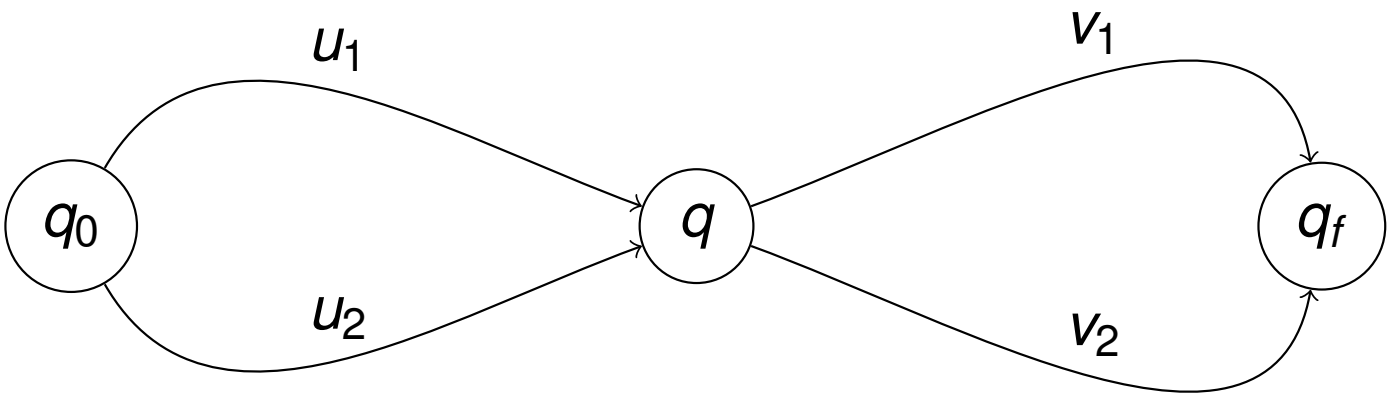
Finite automata



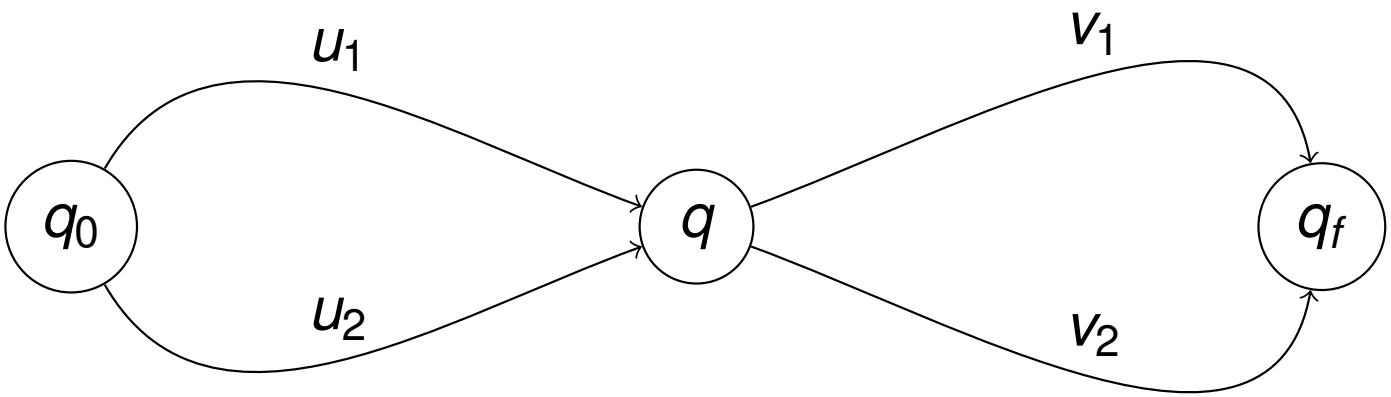
Finite automata



Finite automata



Finite automata



$$\{u_1, u_2, \ldots\} \circ \{v_1, v_2, \ldots\} \subseteq L$$

Decompositions

For a state q

$P_q \circ S_q \subseteq L$

Decompositions

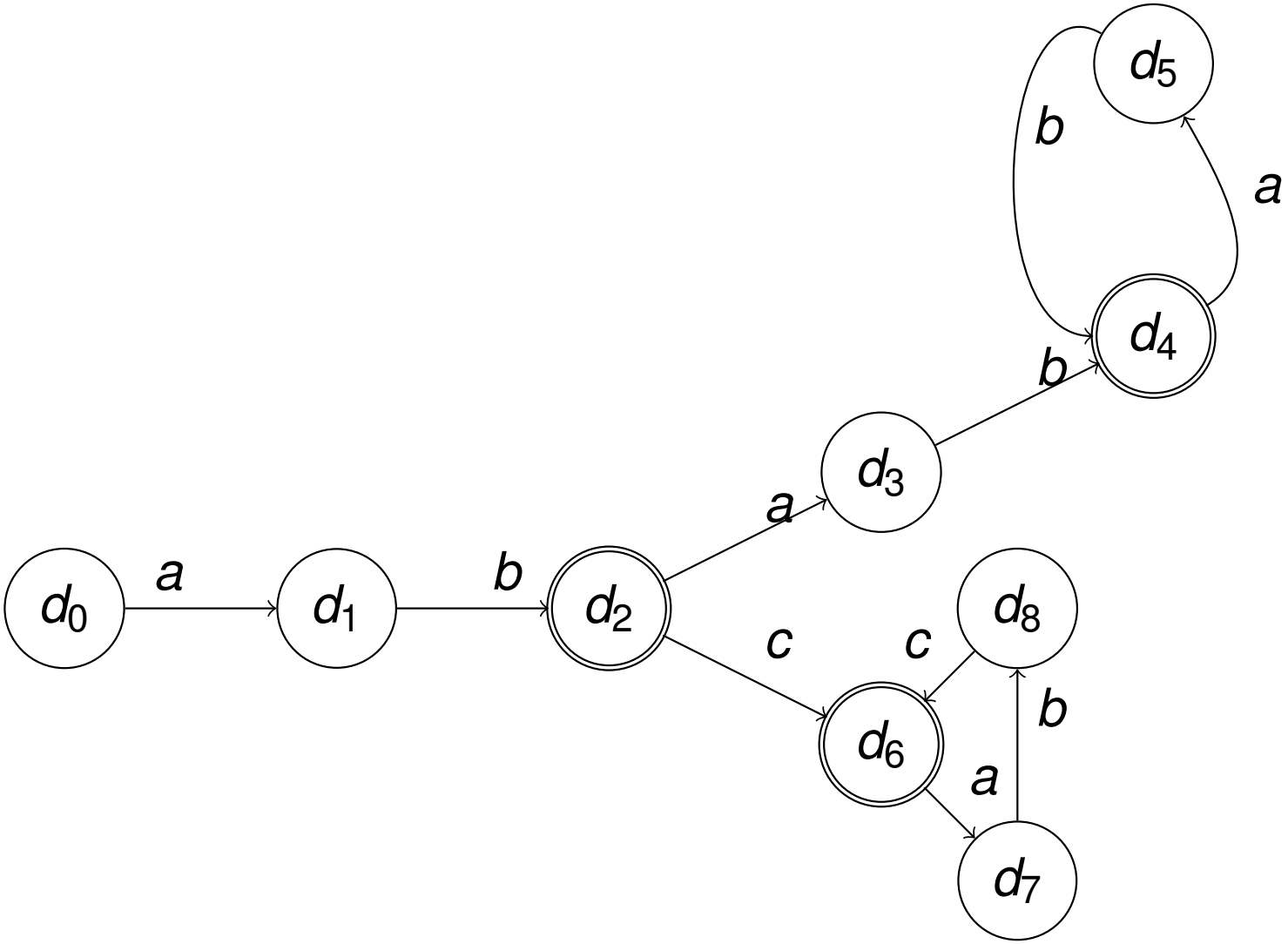
For a state q

$P_q \circ S_q \subseteq L$

- If $\lambda \in P_q$ then it is a start state
- If $\lambda \in S_q$ then it is an end state
- If $u \in P_q$ and $ua \in P_{q'}$ then there is a transition $q \rightarrow^a q'$

DFA

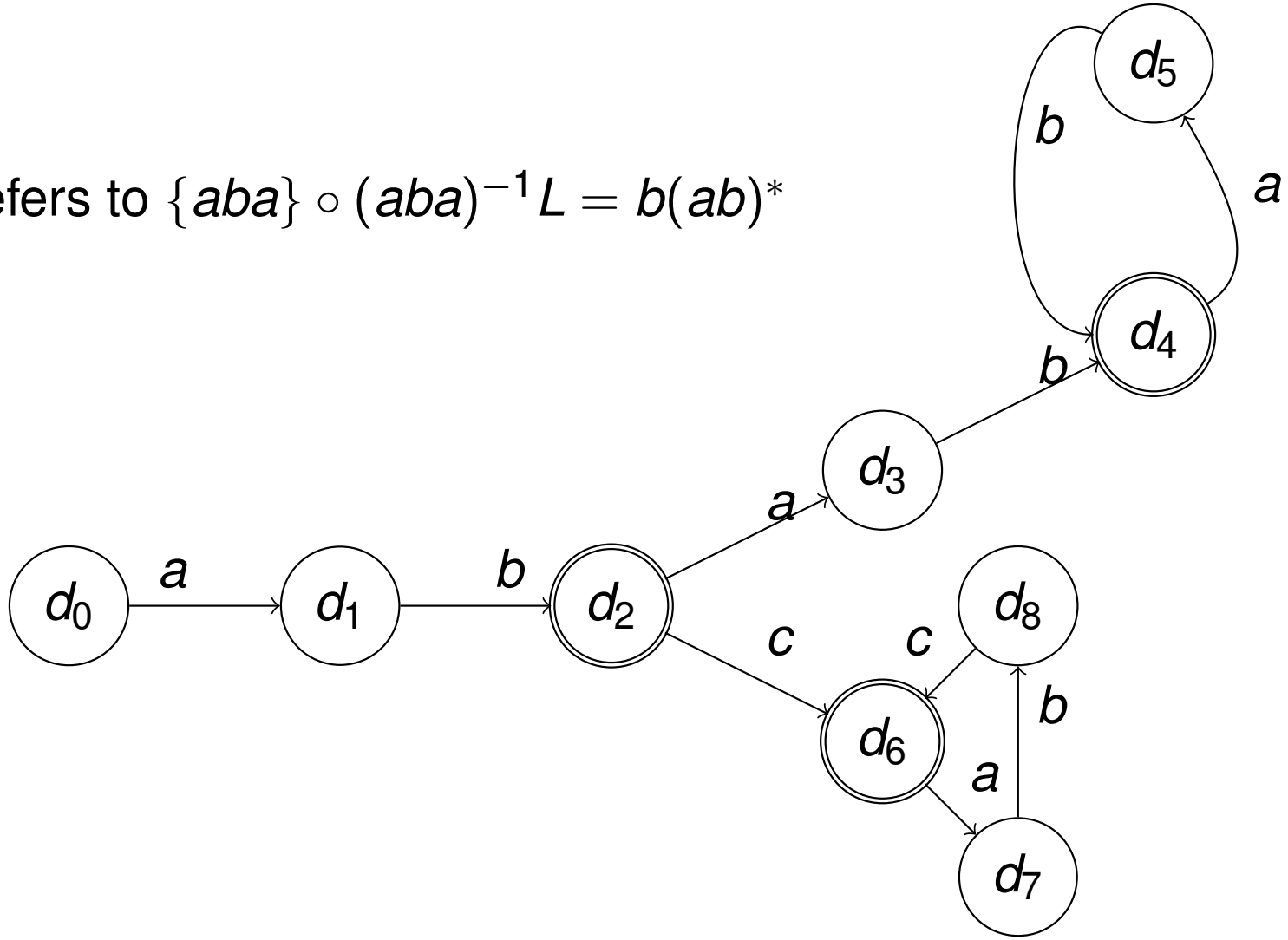
$(ab)^+ \cup (abc)^+$



DFA

$(ab)^+ \cup (abc)^+$

d_3 refers to $\{aba\} \circ (aba)^{-1}L = b(ab)^*$



Myhill-Nerode theorem

$$u^{-1}L = \{v \mid uv \in L\}$$

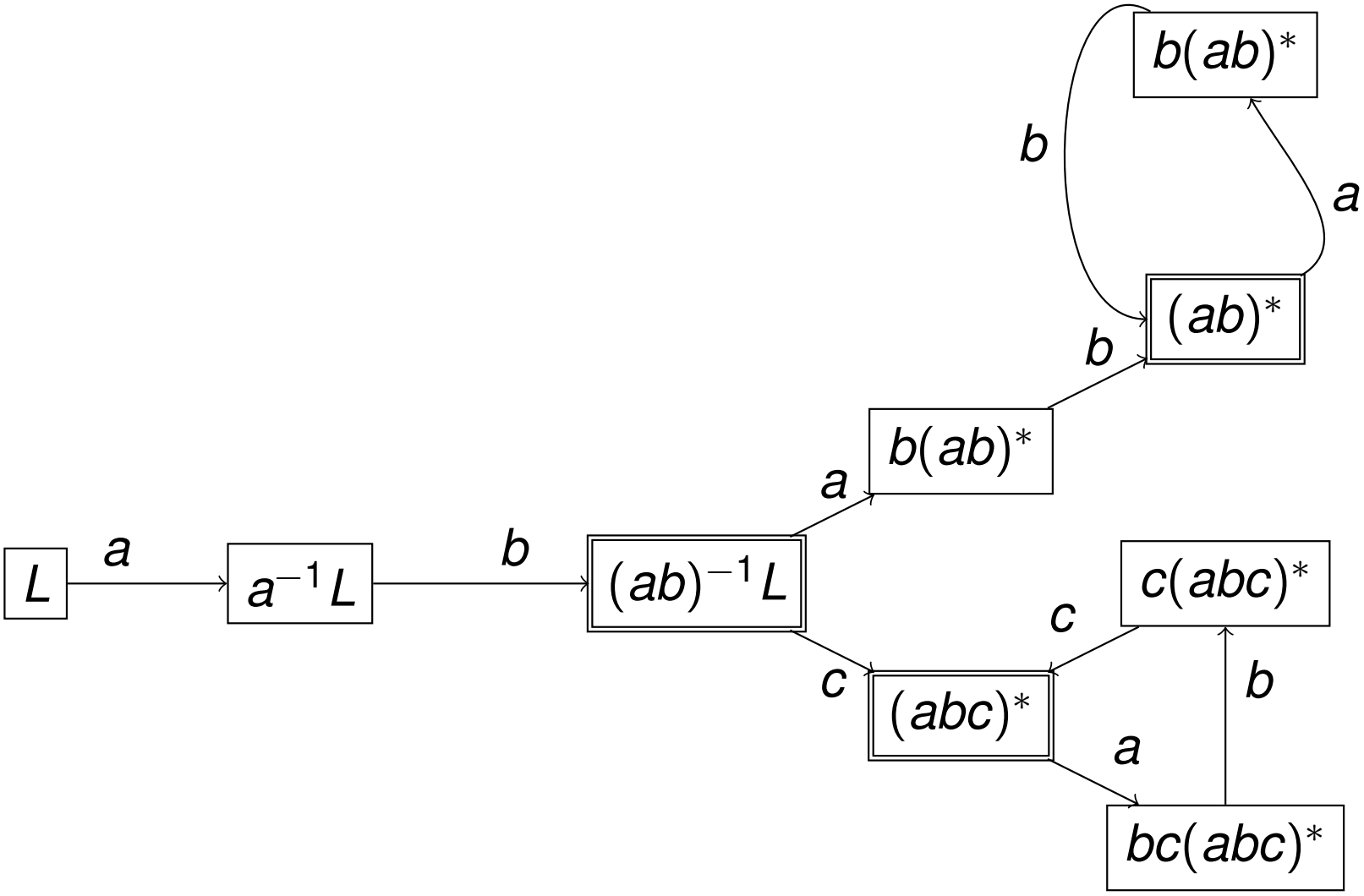
$$w_1 \sim_L w_2 \text{ iff } w_1^{-1}L = w_2^{-1}L$$

$$[u]_{\sim_L} = \{v \mid u^{-1}L = v^{-1}L\}$$

The number of equivalence classes of $\sim_L$ is equal to the number of states in the minimal DFA that accepts L.

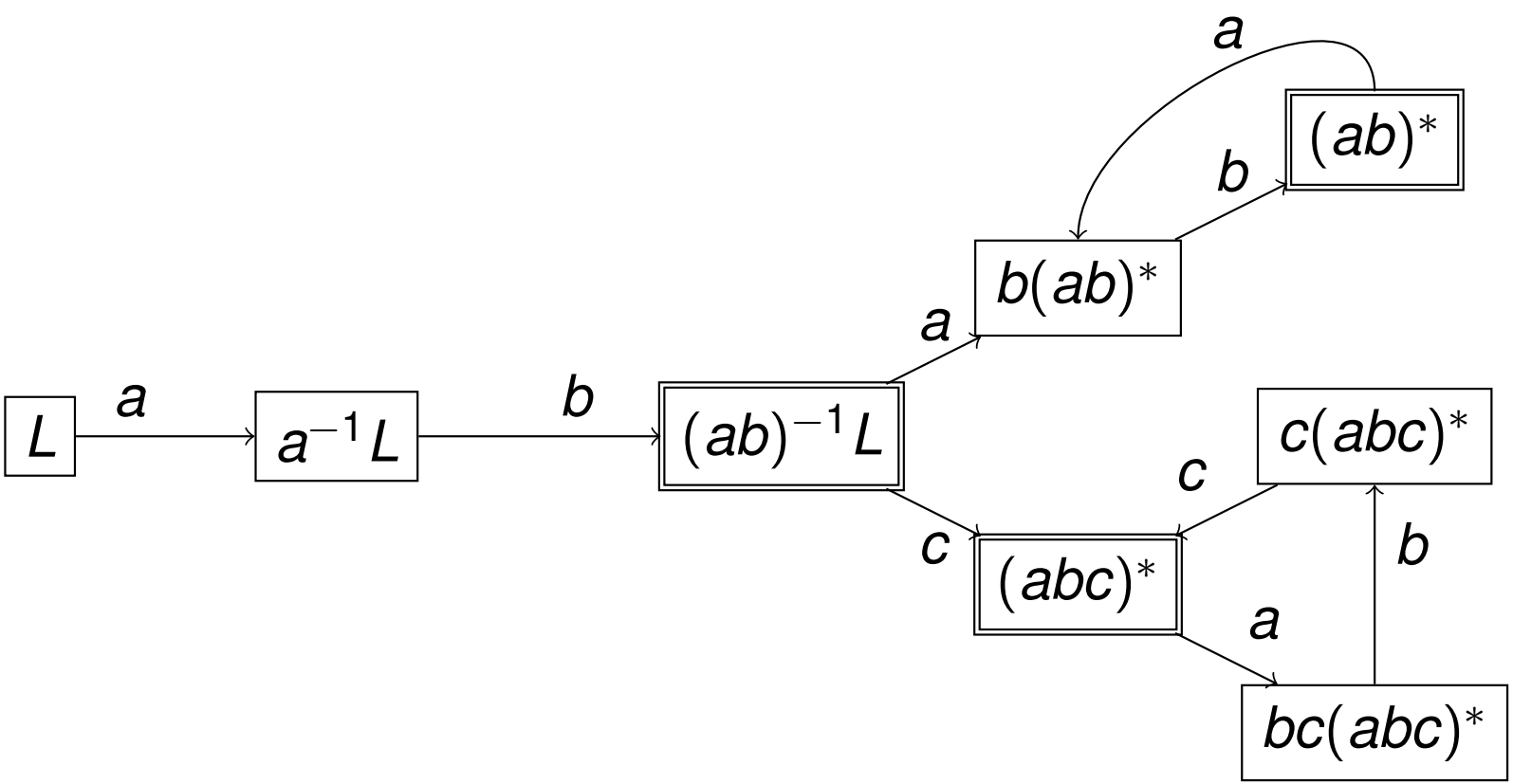
Minimal DFA

$(ab)^+ \cup (abc)^+$



Minimal DFA

$(ab)^+ \cup (abc)^+$



Nice properties of the Canonical DFA

Minimality

It is the smallest DFA that generates L.

Nice properties of the Canonical DFA

Minimality

It is the smallest DFA that generates L.

Universality or “Maximality”

It is (contains) an exact morphic image of every DFA that generates L.

Morphisms of automata

Definition

A morphism from an automaton A to an automaton B is a function:
ϕ from Q_A to Q_B such that: for all $q \in Q_A$:

$$\phi(q_0^A) = q_0^B$$

$$\phi(F_A) \subseteq F_B$$

where defined $\delta_B(\phi(q), a) = \phi(\delta_A(q, a))$

Morphisms of automata

Definition

A morphism from an automaton A to an automaton B is a function:
ϕ from Q_A to Q_B such that: for all $q \in Q_A$:

$$\phi(q_0^A) = q_0^B$$

$$\phi(F_A) \subseteq F_B$$

where defined $\delta_B(\phi(q), a) = \phi(\delta_A(q, a))$

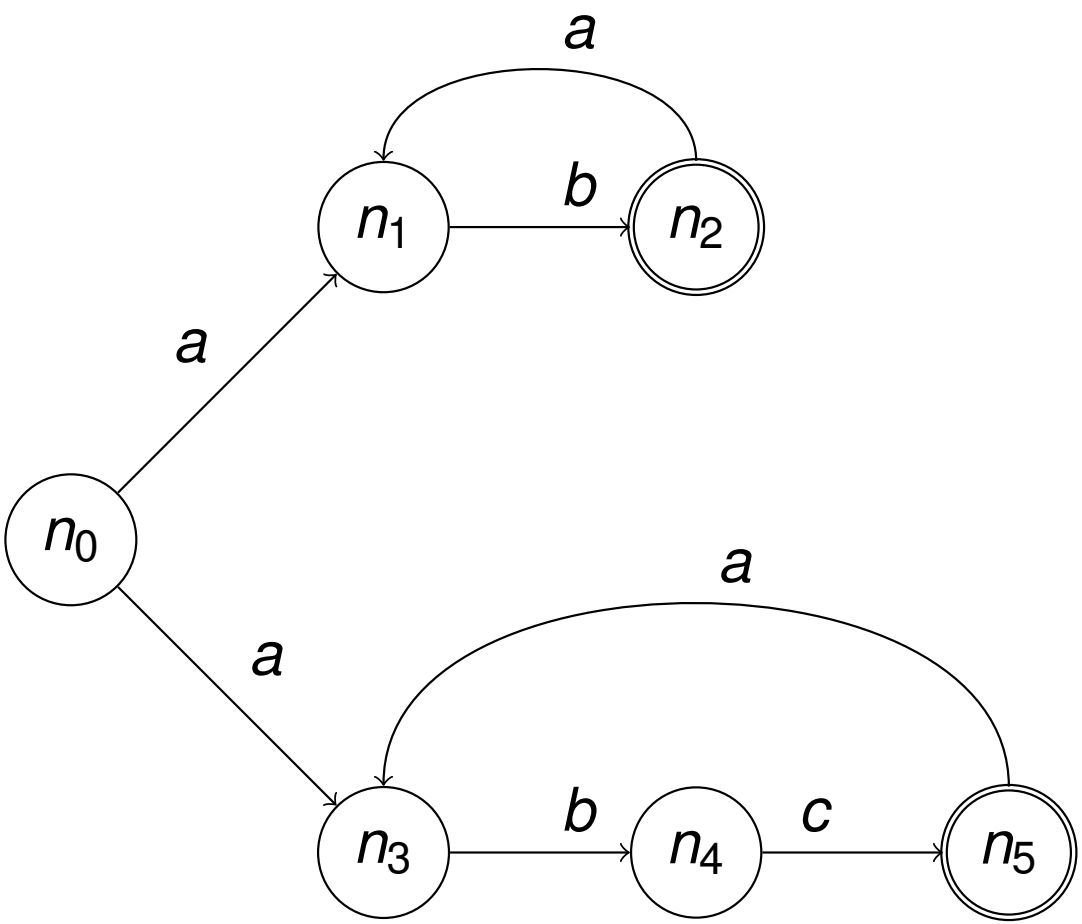
The morphism merges states p, q when $\phi(p) = \phi(q)$.

Morphism into the canonical DFA

- The canonical DFA has states which are decompositions of the form $[u]_{\sim_L} \circ u^{-1}L$.
- So it only has one state for each residual language and because the equivalence classes $[u]$ are a partition this is deterministic.
- Other DFAs will have states $P \circ u^{-1}L$ where $P \subset [u]$,
- Canonical morphism ϕ_L maps each state in a DFA to the unique state in the canonical DFA.

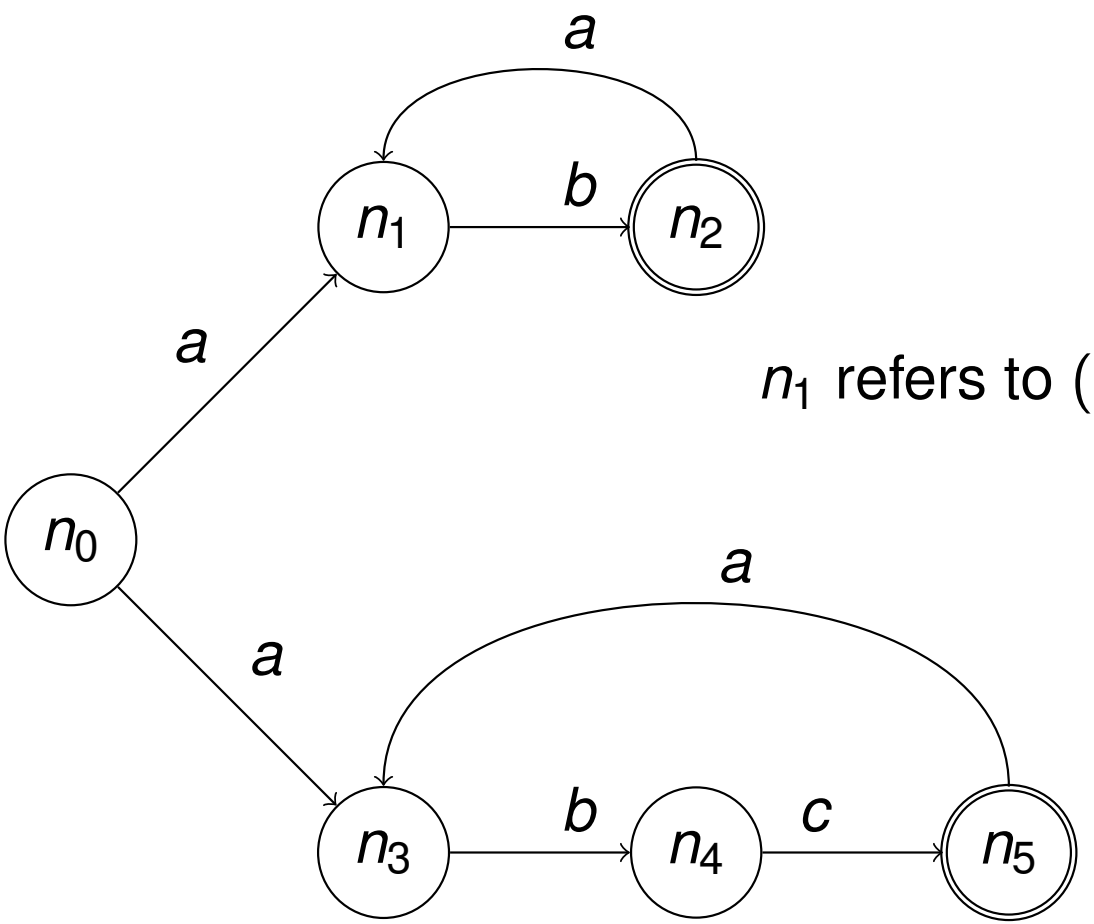
NFA

$(ab)^+ \cup (abc)^+$



NFA

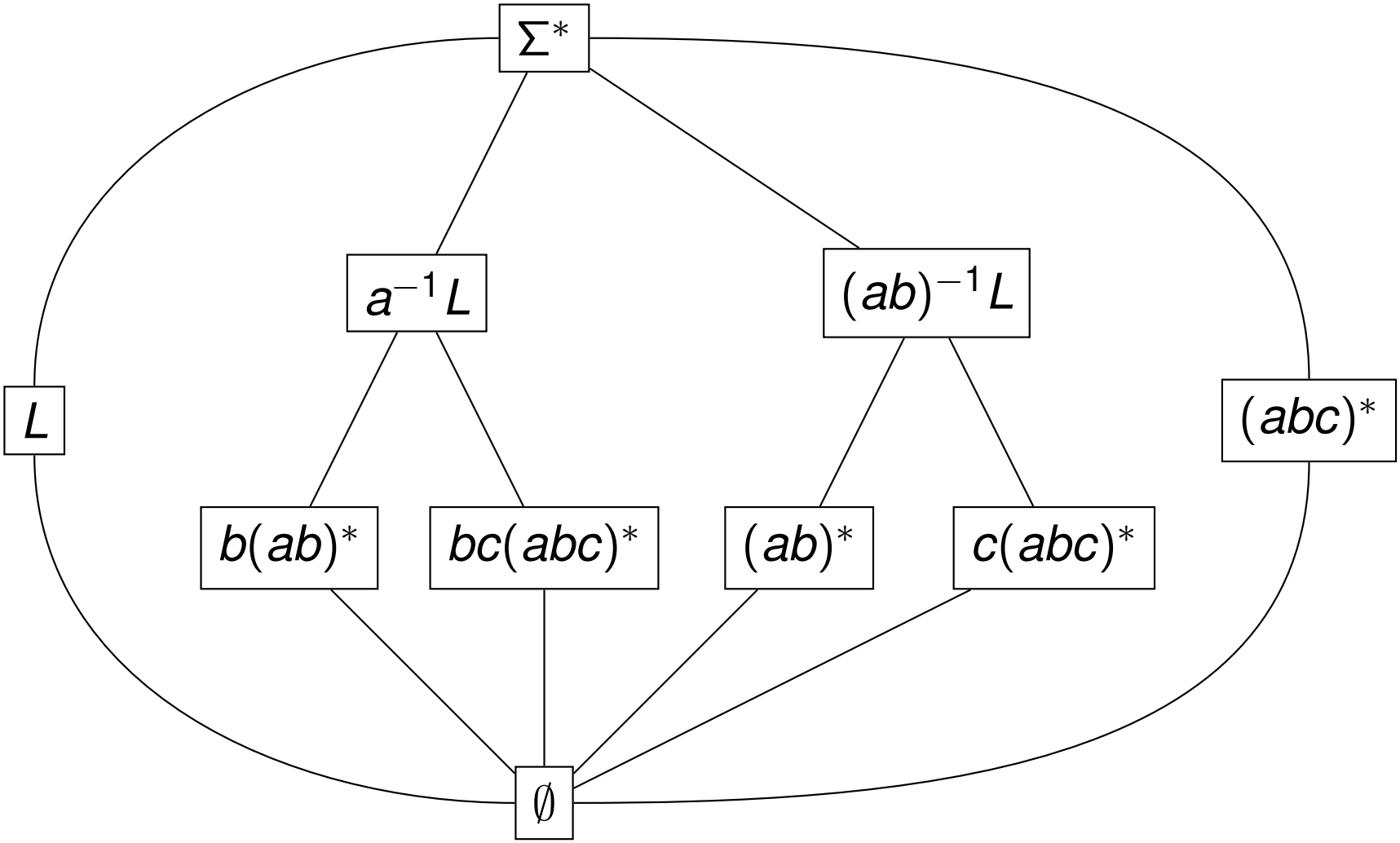
$(ab)^+ \cup (abc)^+$



n_1 refers to $((ab)^* a) \circ (b(ab)^*)$

A hierarchy of maximum factorisations

Showing suffixes



Universal automaton

States

Maximal factorisations P, S such that $PS \subseteq L$.

Start states

States such that $\lambda \in P$.

Accepting states

States such that $\lambda \in S$

Transitions

Transition from $\langle P, S \rangle \rightarrow^a \langle P', S' \rangle$ iff $Pa' \subseteq P'$

Nice properties of the UA for a regular language L

Universality

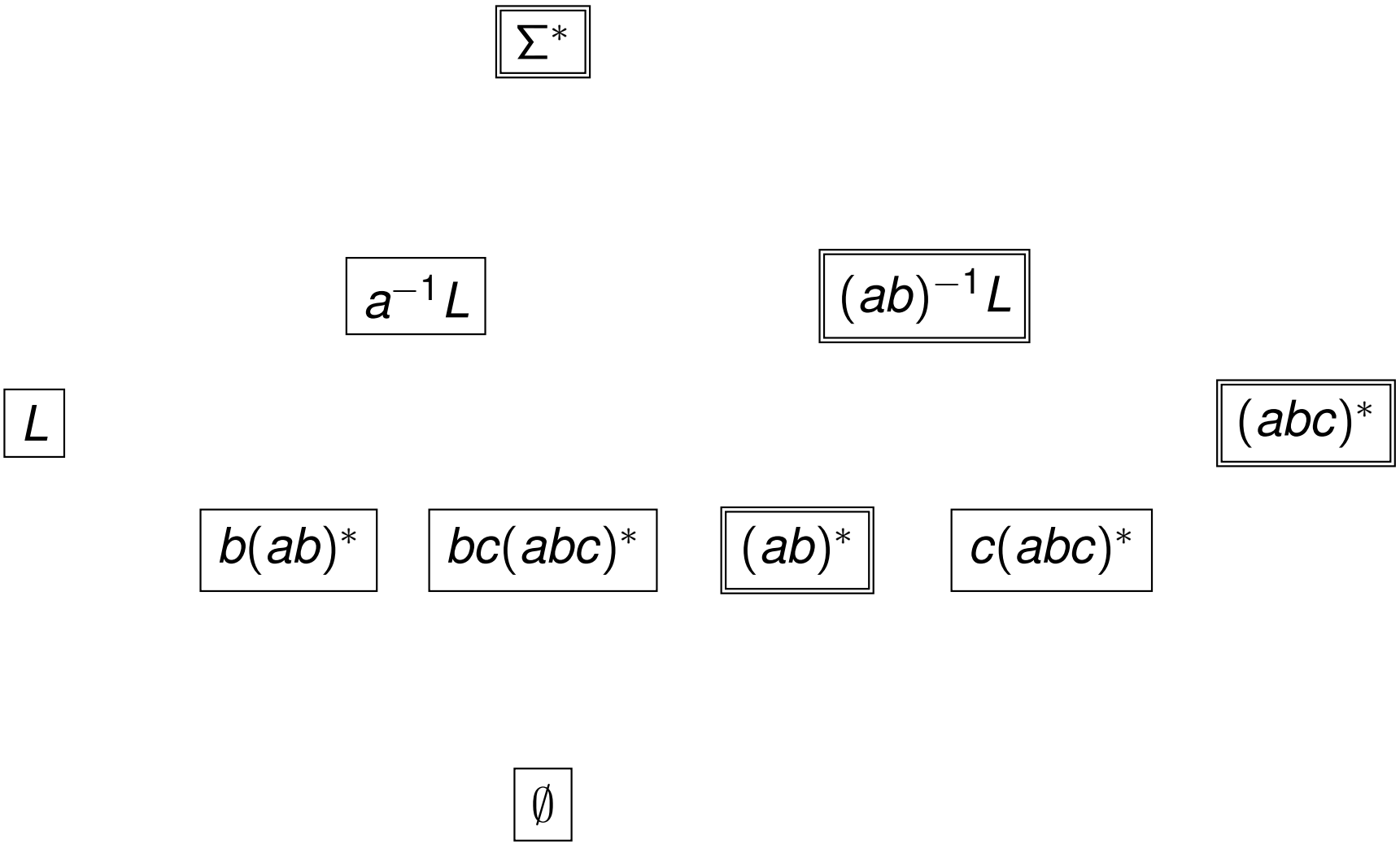
It contains a morphic image of every FA that generates L.

Minimality

It contains every minimal automaton as a subautomaton.

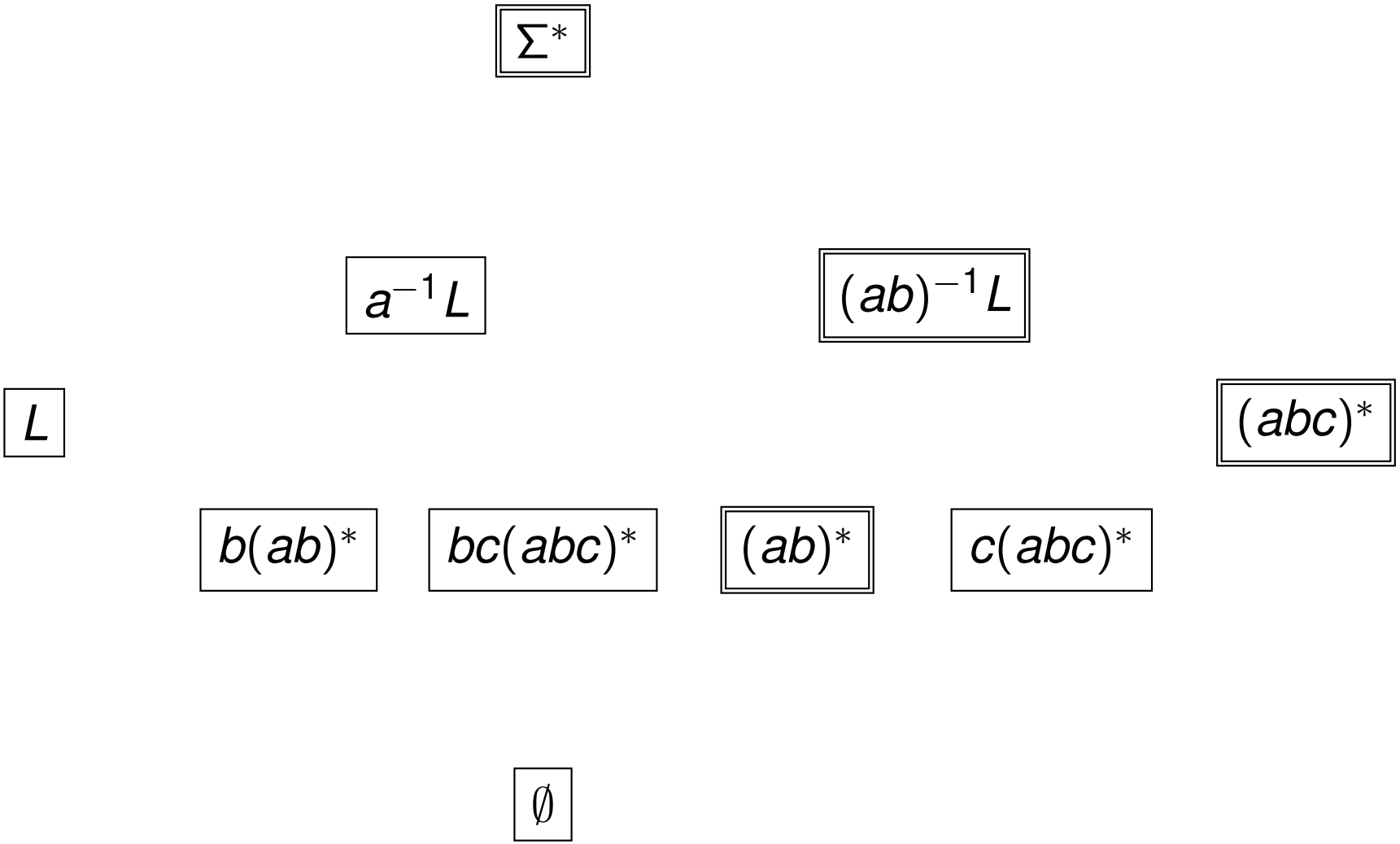
Hierarchy of states

$(ab)^+ \cup (abc)^+$



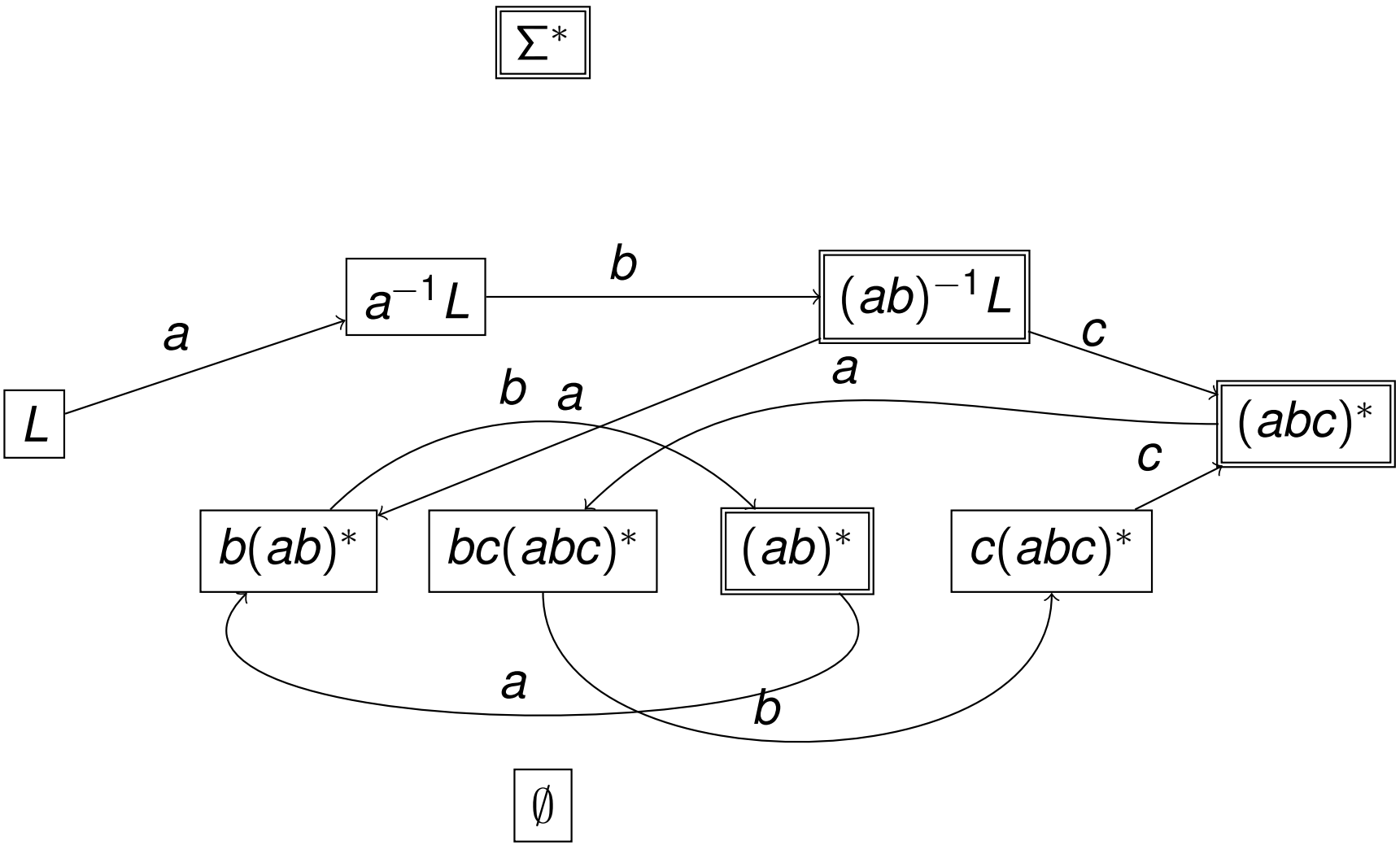
Universal automaton

$(ab)^+ \cup (abc)^+$



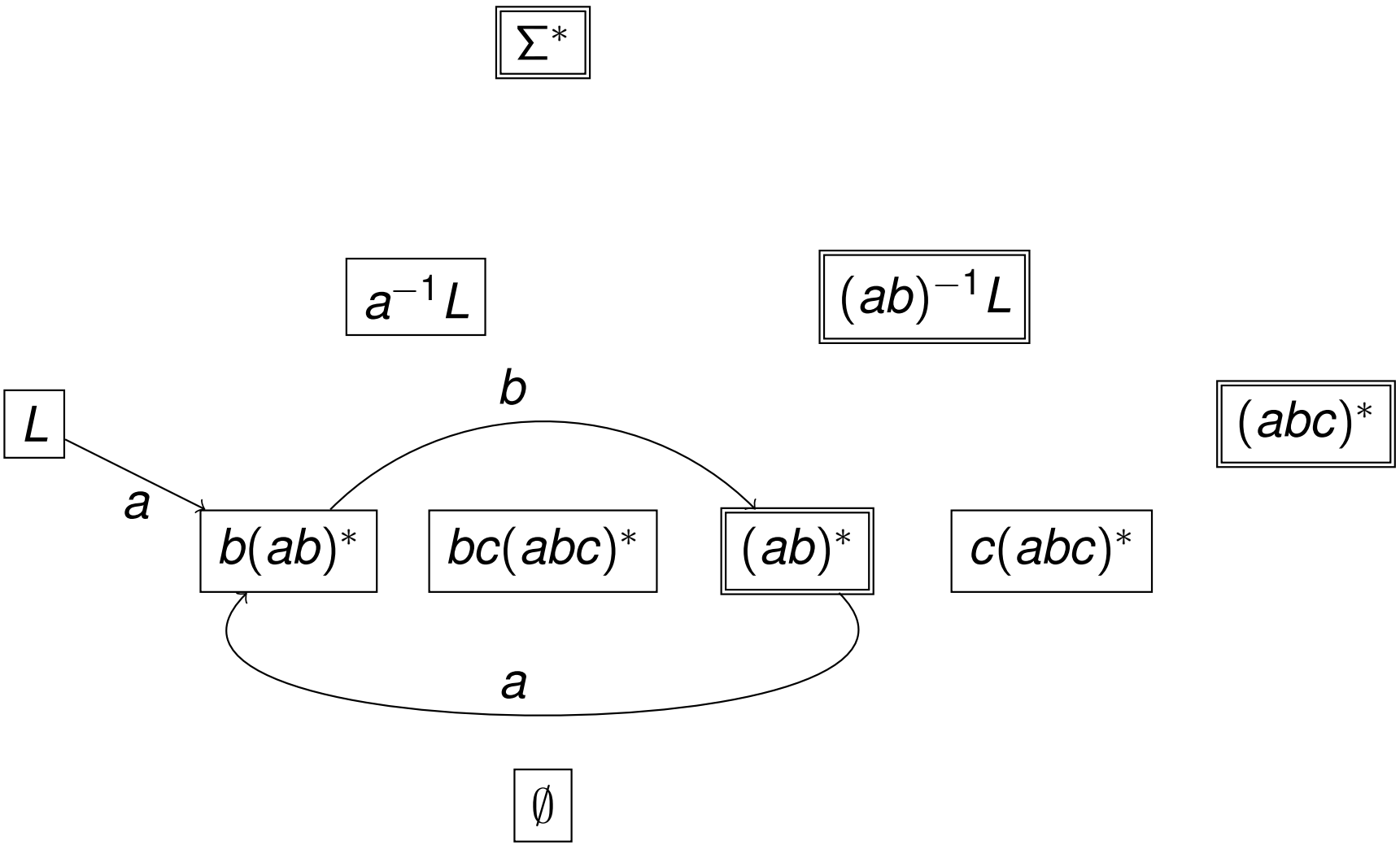
Universal automaton

$(ab)^+ \cup (abc)^+$



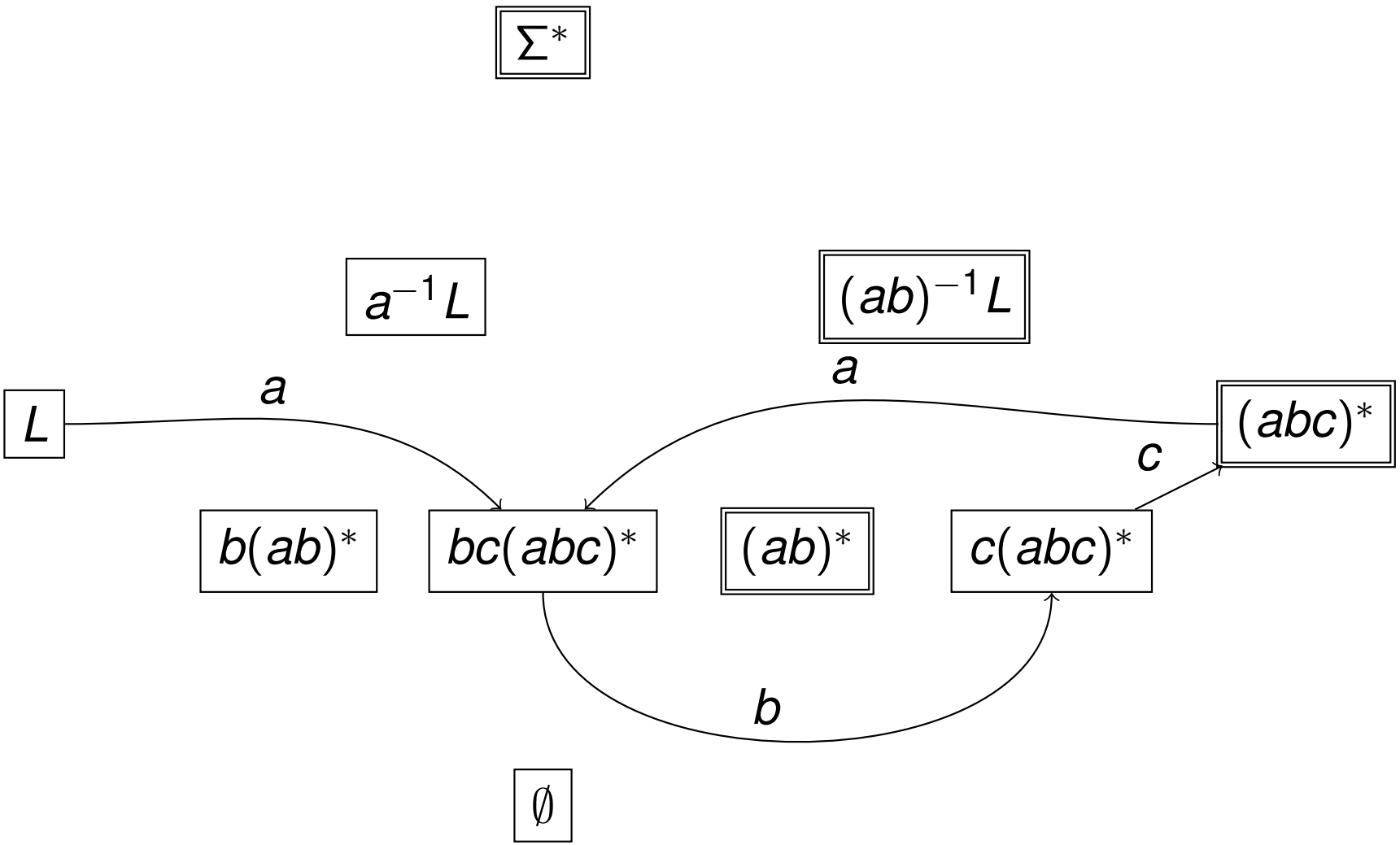
Universal automaton

$(ab)^+ \cup (abc)^+$



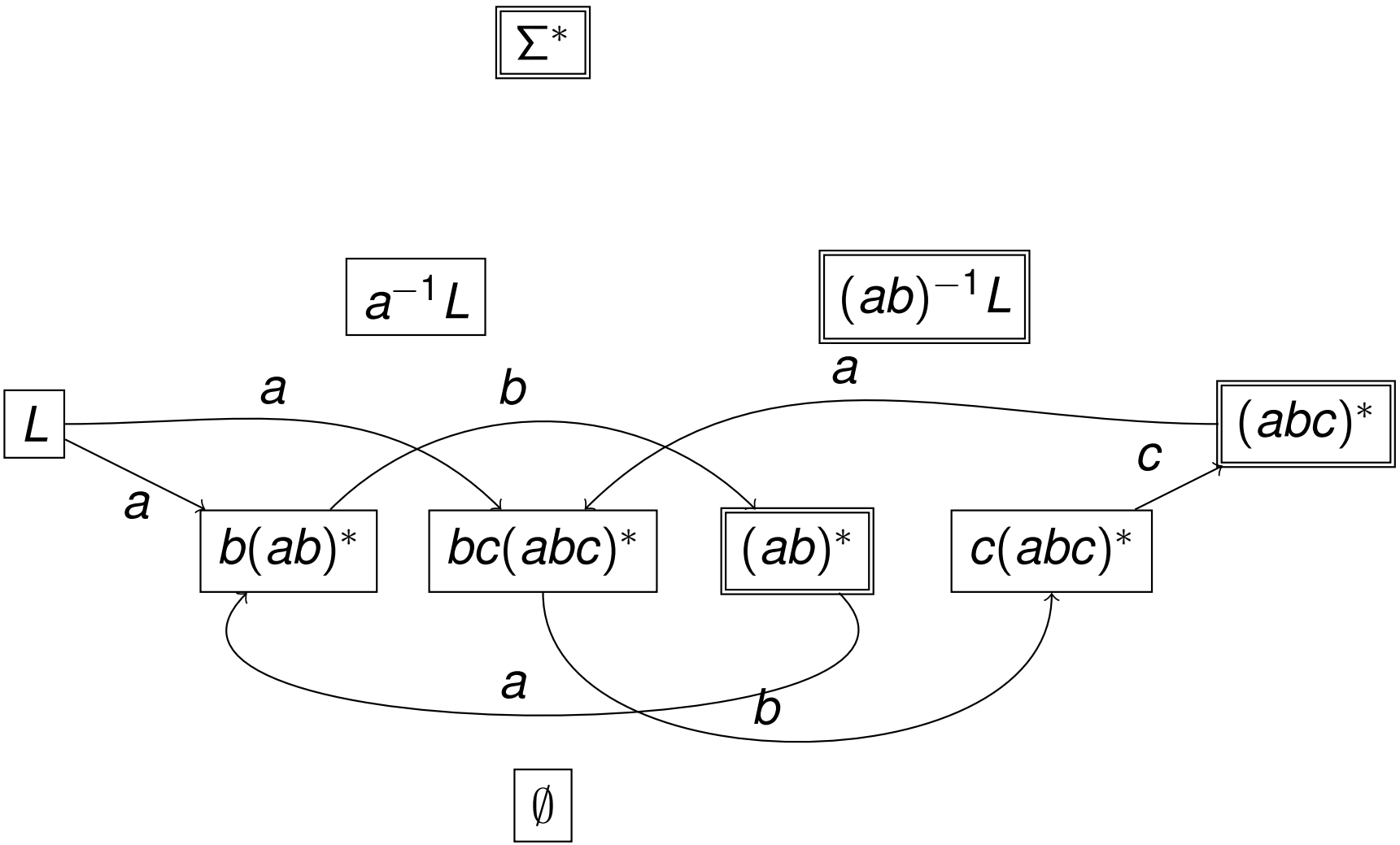
Universal automaton

$(ab)^+ \cup (abc)^+$



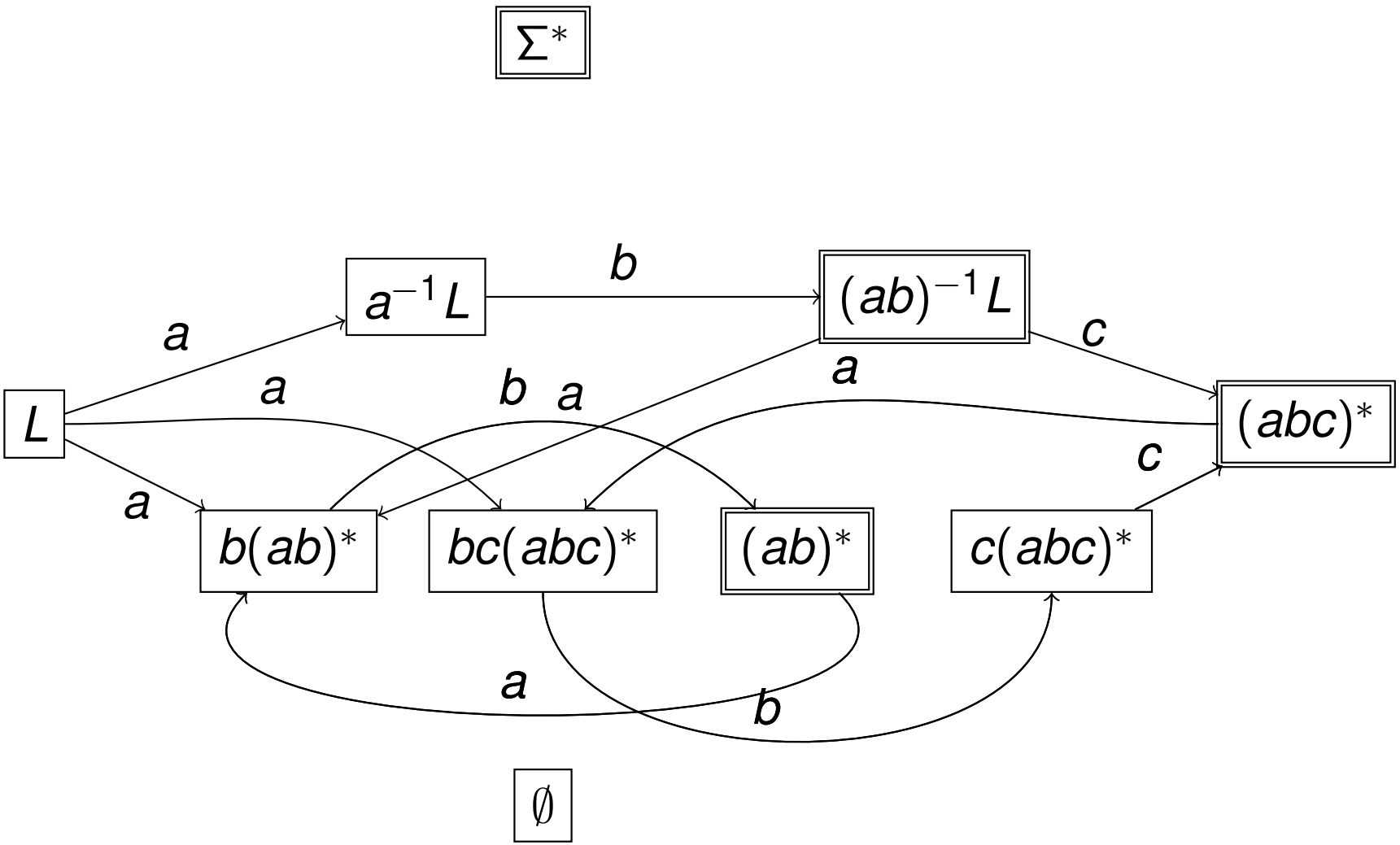
Universal automaton

$(ab)^+ \cup (abc)^+$



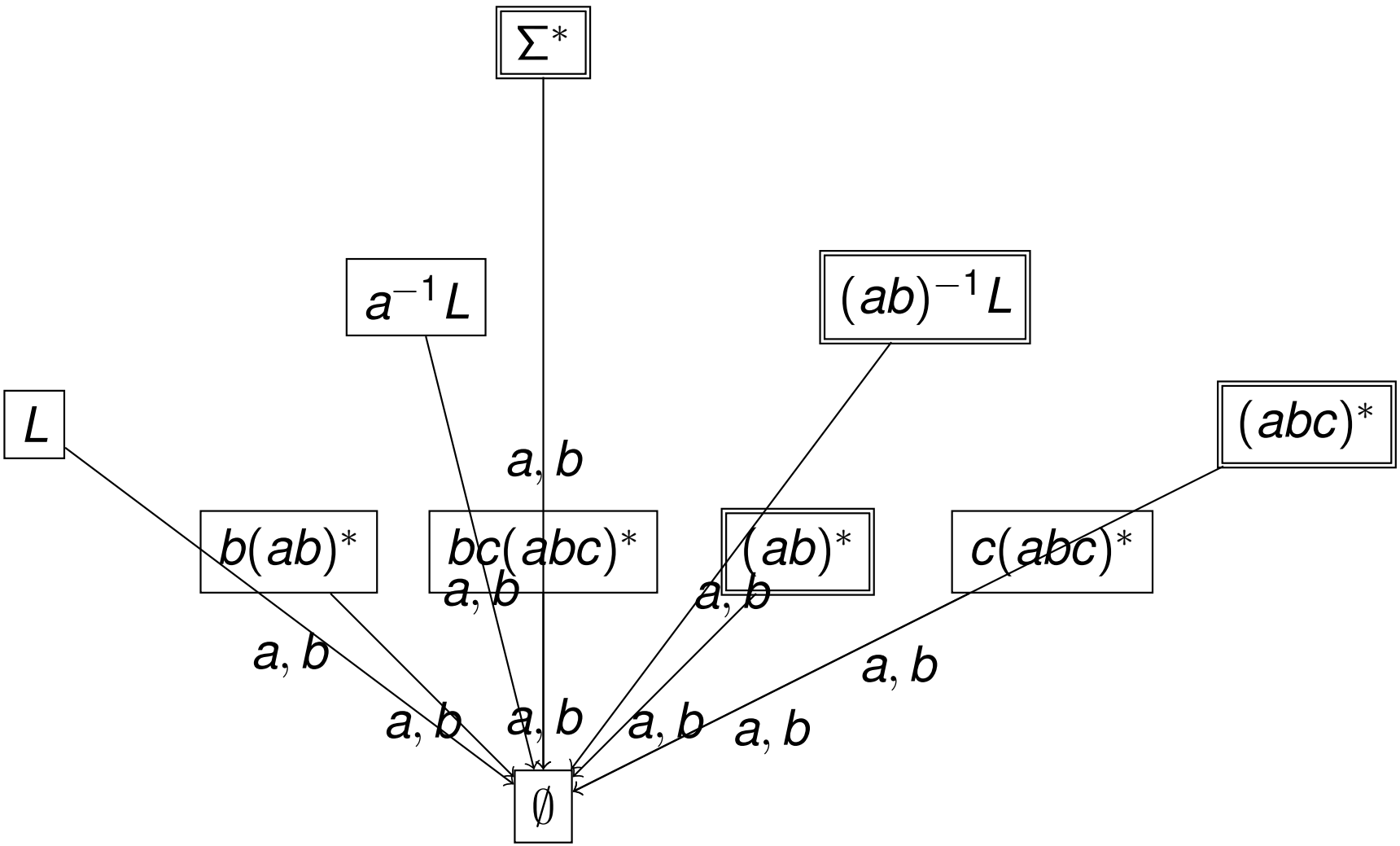
Universal automaton

$(ab)^+ \cup (abc)^+$



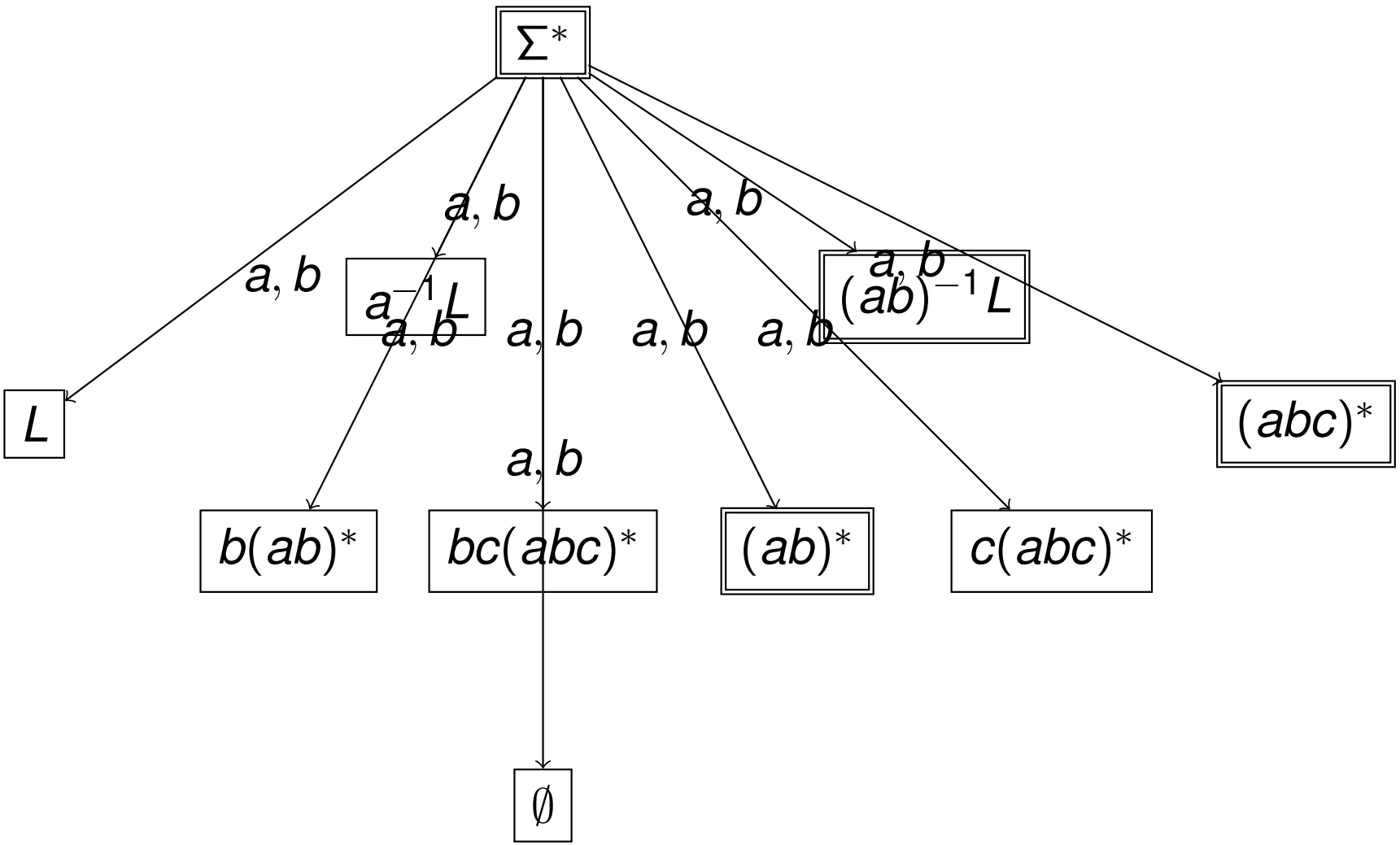
Universal automaton

$(ab)^+ \cup (abc)^+$



Universal automaton

$(ab)^+ \cup (abc)^+$



Regular languages

Question

What does the state in a DFA mean?
It is a residual language.

Question

What is the state in a regular grammar/NFA?
It is a maximal decomposition $P \circ Q \subseteq L$.

Regular languages

Question

What does the state in a DFA mean?
It is a residual language.

Question

What is the state in a regular grammar/NFA?
It is a maximal decomposition $P \circ Q \subseteq L$.

States are decompositions

An automaton defines a decomposition of a particular type.
We can start with a *maximal* decomposition, and end with a *minimal* automaton.

Outline

Definitions

Context-free grammar

A tuple $\langle \Sigma, V, S, P \rangle$, where

- Σ is a nonempty finite set of terminal symbols
- V is a finite set of nonterminal symbols disjoint from Σ
- $S \in V$ is the start symbol
- $P \subset V \times (V \cup \Sigma)^+$ is a set of productions written $N \to \alpha$

Questions

What are the elements of $V = \{N, P, Q \dots\}$? What do they mean? What does S mean? What does a production like $N \to aPQ$ mean?

Semantics of Context-free languages I

Rewriting systems (Chomsky, 1959)

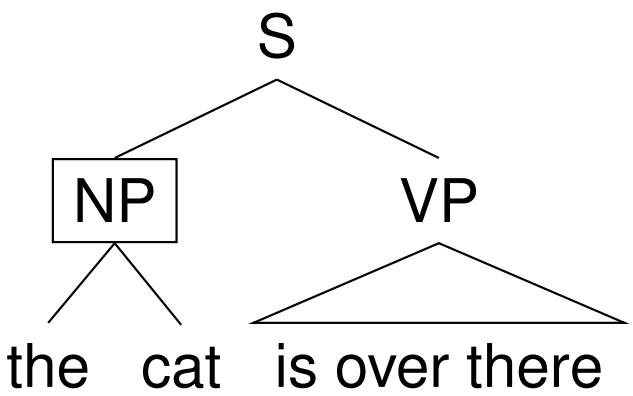
$N \to aPQ$

$\alpha N \beta \Rightarrow \alpha aPQ \beta$

$S \stackrel{*}{\Rightarrow} abcaabb.$

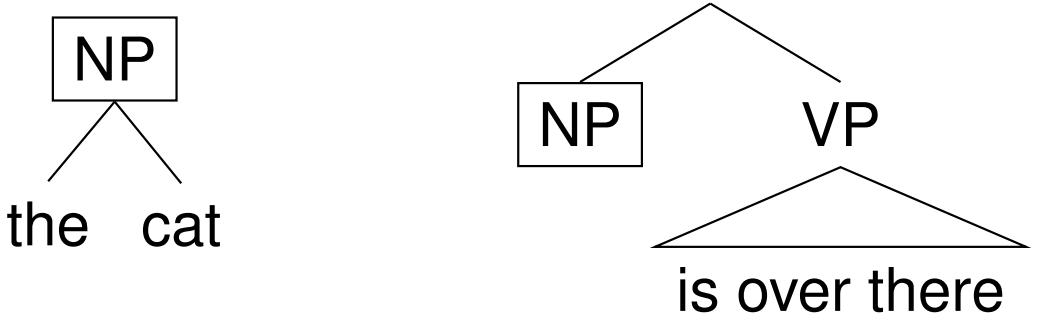
Semantics of Context-free languages II

Trees: (Thatcher, 1967)



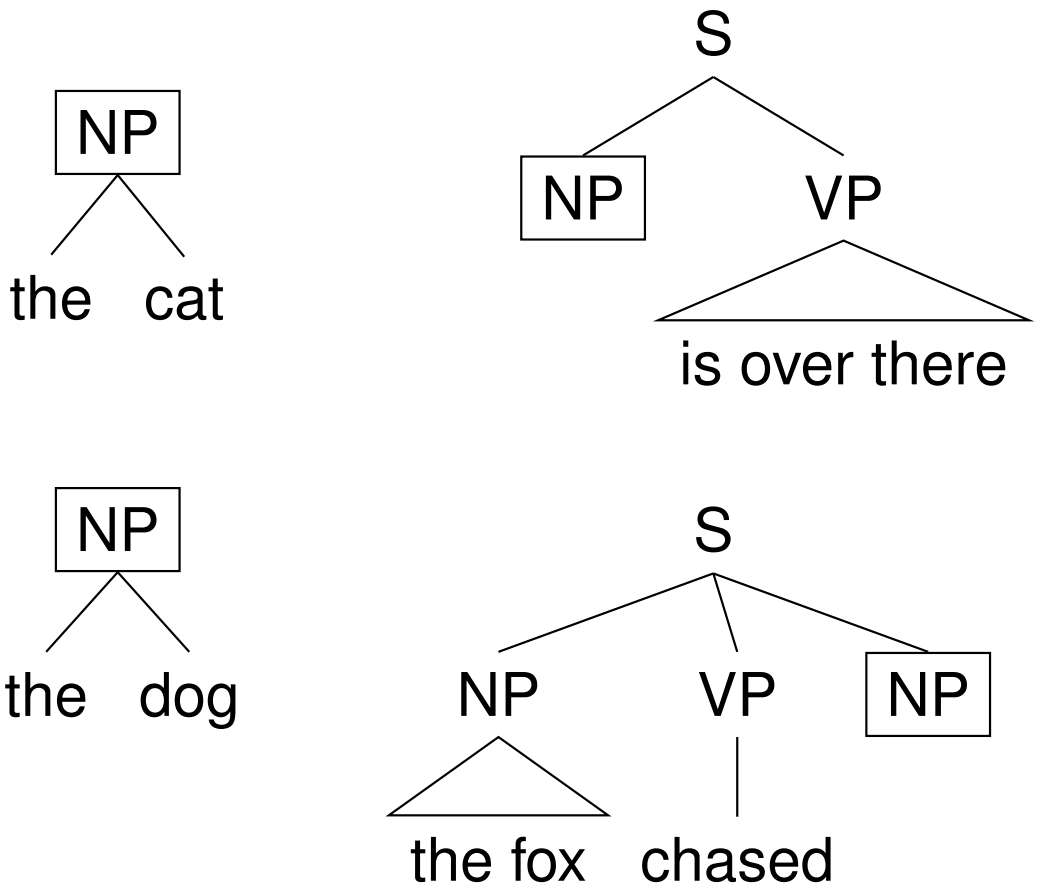
Semantics of Context-free languages II

Trees: (Thatcher, 1967)



Semantics of Context-free languages II

Trees: (Thatcher, 1967)



Definition

Contexts

A context is a string with a hole: $l \square r$.

Filling the hole

$l \square r \odot u = lur$

A factorisation

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

$$C \odot S \subseteq L$$

Context free grammars

Contexts and yields

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

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

Congruential CFGs

Analogous to DFAs

- Nonterminals correspond to equivalence classes of strings that occur in the same set of contexts: the syntactic congruence.

Definition

Define $u^{\triangleright} = \{l \square r \mid lur \in L\}$
$u \equiv_L v$ iff $u^{\triangleright} = v^{\triangleright}$.
$[u]_L = \{v \mid u^{\triangleright} = v^{\triangleright}\}$

This gives us grammars with a “bottom-up” determinism.

Congruential CFGs

- If the language is not regular there will be an infinite number of congruence classes, but we don't need all the classes.
- Not all context-free languages can be represented with a congruential CFG: e.g. the palindrome language $\{w \mid w = w^R\}$ has congruence classes which are just single strings.
- These grammars can be efficiently learned in the exact query model.
- Like DFAs, parsing speed does not depend on the size of the grammar.
- Closely related to the NTS languages, which can be parsed in linear time.

Semantics of Context-free languages III

Equations: (Ginsburg and Rice, 1962)

Ginsburg and Rice, 1962

$S \to ab, S \to aSb, S \to \epsilon$

Interpret this as a set of equations:

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

The language is the smallest solution to these equations.

The variables take values in $\mathcal{P}(\Sigma^*)$.

Semirings

What is $\mathcal{P}(\Sigma^*)$?

Free structure over Σ with

- One associative operation $\circ$
- A unit λ
- A countable union operation $\vee$ that is associative and idempotent
- A unit with respect to $\vee$
- $\vee$ distributes over $\circ$, ...

Semirings

What is $\mathcal{P}(\Sigma^*)$?

Free structure over Σ with

- One associative operation $\circ$
- A unit λ
- A countable union operation $\vee$ that is associative and idempotent
- A unit with respect to $\vee$
- $\vee$ distributes over $\circ$, ...

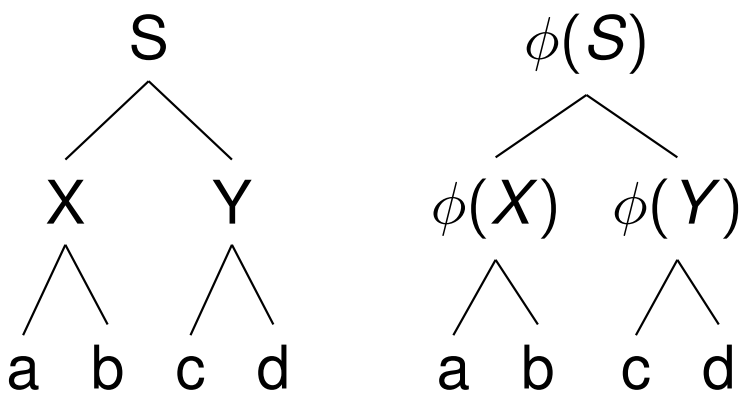
Algebraic structure

Complete idempotent semiring (CIS)

CFG-Morphism

Definition of a grammar morphism

- Suppose we have a CFG $G = \langle \Sigma, V, S, P \rangle$ and a function $\phi : V \to X$.
- Then $\phi(G)$ is the grammar $\langle \Sigma, \phi(V), \phi(S), \phi(P) \rangle$
- If $(N \to aXY) \in P$ then $(\phi(N) \to a\phi(X)\phi(Y)) \in \phi(P)$ so $\mathcal{L}(G) \subseteq \mathcal{L}(\phi(G))$



Merging

S

C

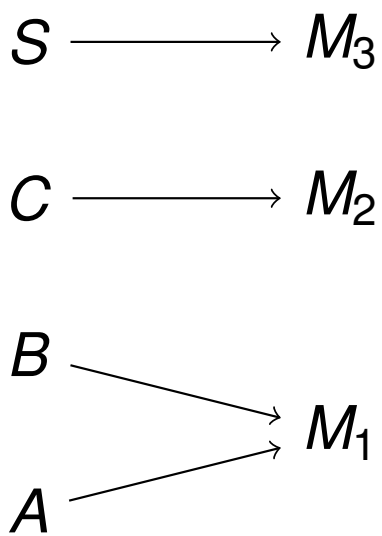
B

A

Grammar

$S \to ASB$, $S \to C$, $A \to a$, $B \to b$, $C \to c$

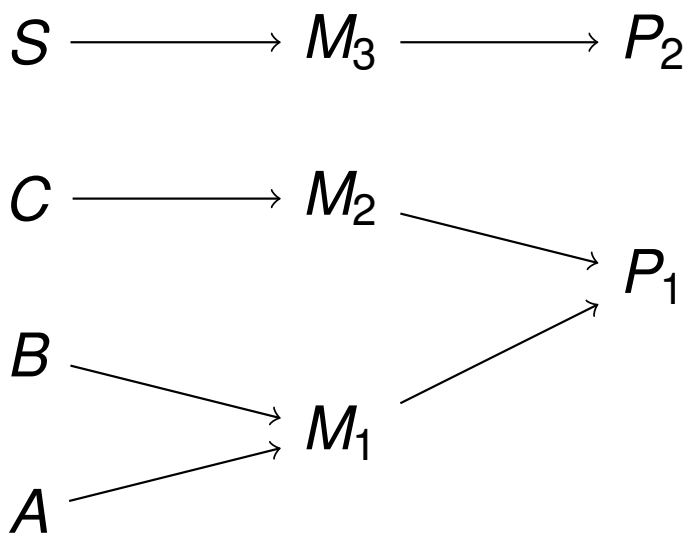
Merging



Merged grammar

$M_3 \to M_1 M_3 M_1$, $M_3 \to M_1$, $M_1 \to a$, $M_1 \to b$, $M_2 \to c$

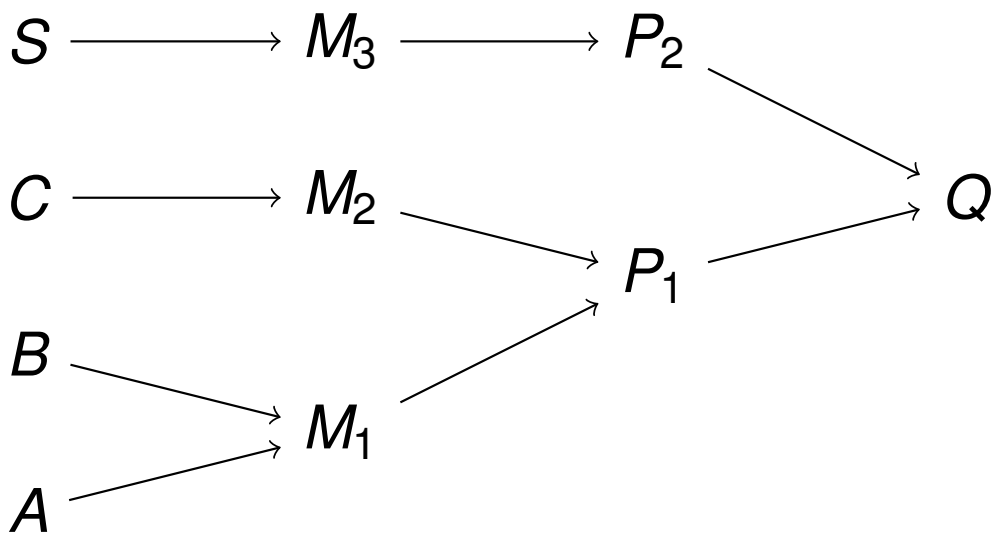
Merging



Merged grammar

$P_2 \to P_1 P_2 P_1$, $P_2 \to P_1$, $P_1 \to a$, $P_1 \to b$, $P_2 \to c$

Merging



Merged grammar

$Q \to QQQ$, $Q \to Q$, $Q \to a$, $Q \to b$, $Q \to c$

Mergable nonterminals

Suppose we have two distinct nonterminals that generate the same language:

- A, B such that $\mathcal{L}(G, A) = \mathcal{L}(G, B) = X$.
- Then we can merge them with a map $\phi(A) = \phi(B)$.
- $\phi : N \to \mathcal{L}(G, N)$

Clearly $\mathcal{L}(\phi(G)) = \mathcal{L}(G)$

Homomorphisms give morphisms

Idea

A CIS-homomorphism $f : \mathcal{P}(\Sigma^*) \to A$ gives a CFG-morphism.
$\phi_f(N) = f(\mathcal{L}(G, N))$.
Note that

- $S = (a \circ b) \vee (a \circ S \circ b) \vee \epsilon$
- So $f(S) = (f(a) \circ f(b)) \vee (f(a) \circ f(S) \circ f(b)) \vee f(\epsilon)$

Two extremes

- Identity map just merges nonterminals that generate the same language
- Map to the trivial one-element CIS merges all nonterminals into one.
  This will usually increase the language generated to Σ^*.

Pause

The central idea of the algebraic approach

- We define an algebraic structure – a string concatenation algebra appropriate for our grammar.
- We think of grammars as sets of equations in this algebra.
- Then homomorphisms of the algebra will give morphisms of the grammar.

Recognising

Residual

Define an "inverse" (actually a residual) of f, called f^*.

- $f : A \to B$ is a CIS-homomorphism
- $f^* : B \to A$ is $f(y) = \bigvee \{x \mid f(x) \leq y\}$

Definition

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

Analogue of the classic automata theoretic idea of a monoid recognizing a language

There is a unique answer

We want to merge as much as possible while still generating the same language.

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

There is a unique answer

We want to merge as much as possible while still generating the same language.

Theorem

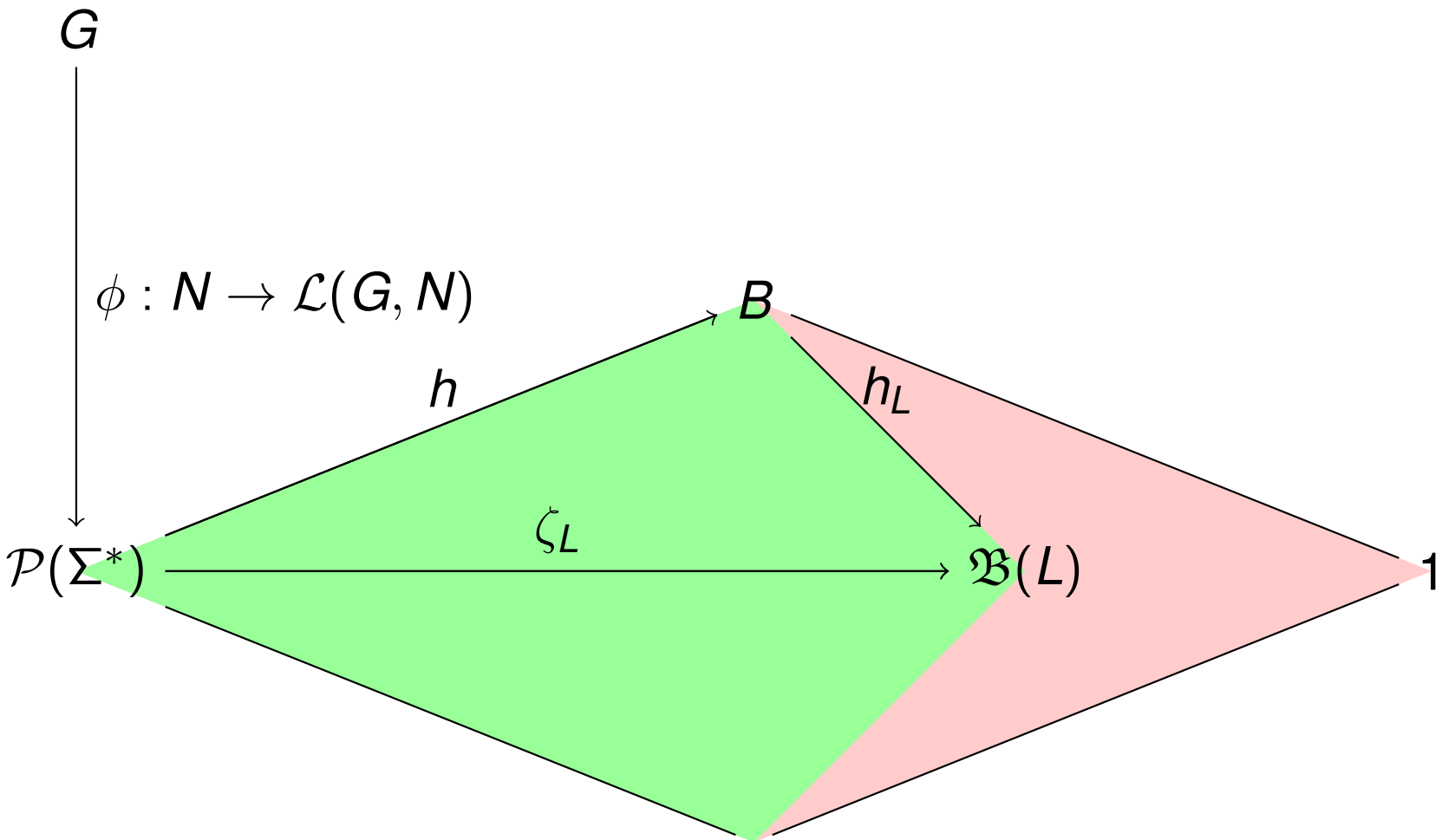
Let G be a CFG over Σ and h a homomorphism $h : \mathcal{P}(\Sigma^*) \to B$. Then

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

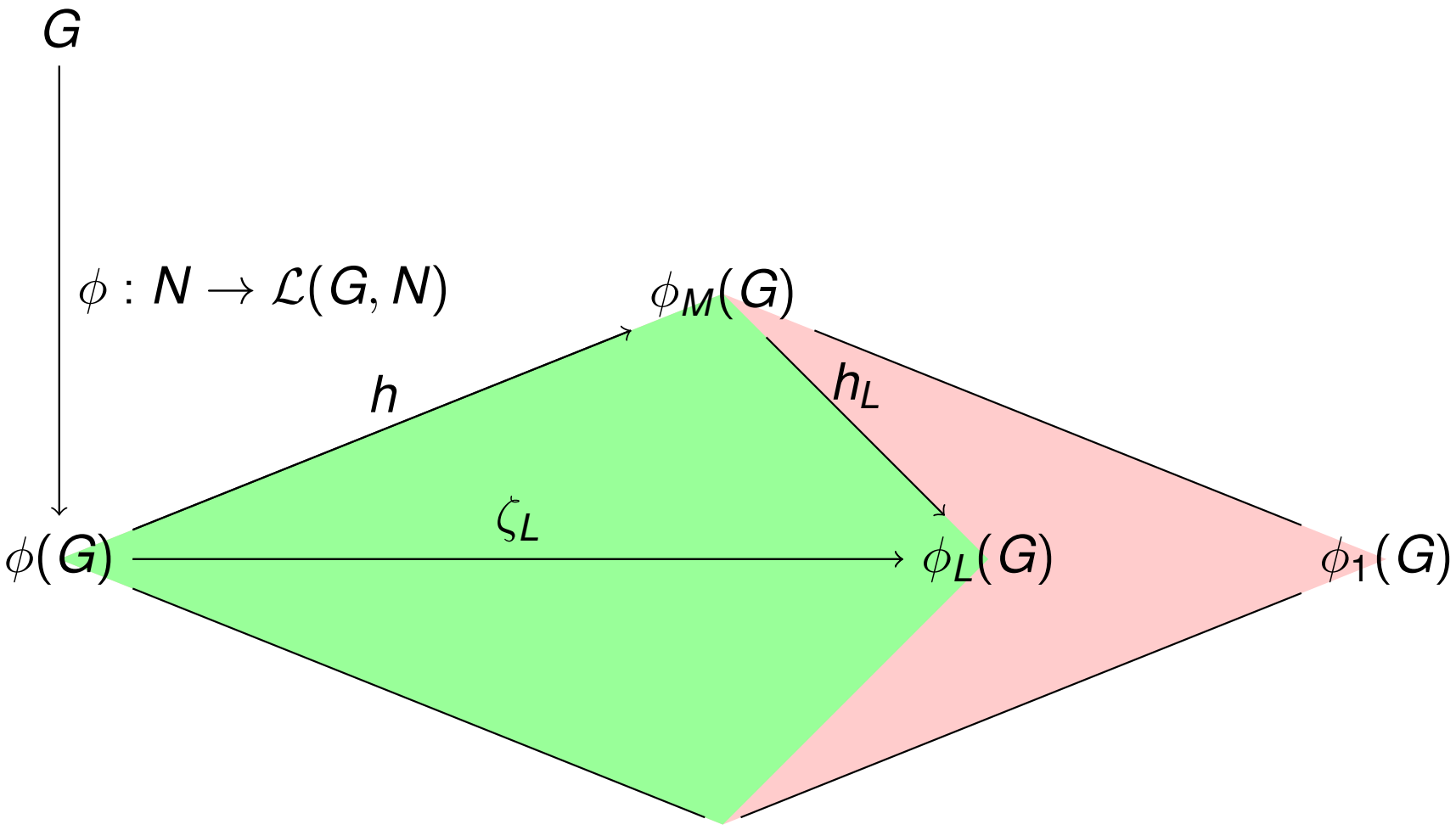
Uniqueness

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

The universal morphism



The universal CFG-morphism



Outline

Maximal decompositions into contexts and substrings

Definition

A *syntactic concept* is a pair $\langle S, C \rangle$ where

- where S is a set of strings, and C is a set of contexts
- $C \odot S \subseteq L$
- C and S are both maximal

Maximal decompositions into contexts and substrings

Definition

A *syntactic concept* is a pair $\langle S, C \rangle$ where

- where S is a set of strings, and C is a set of contexts
- $C \odot S \subseteq L$
- C and S are both maximal

The unique smallest CIS that recognizes L is called the syntactic concept lattice (Clark, 2010), denoted $\mathfrak{B}(L)$, whose elements are these maximal decompositions.

The syntactic concept lattice

If S is a set of strings:

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

If C is a set of contexts:

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

Closed sets of strings

definition

A set of strings is closed if $X = X^{\triangleright\triangleleft}$.
$X^{\triangleright\triangleleft}$ is always closed.

Nonterminals will correspond to closed sets of strings.

Example

Suppose we have a single word $\{w\}$.
$\{w\}^{\triangleright\triangleleft}$ is the set of all words that can occur anywhere w can.

Implications

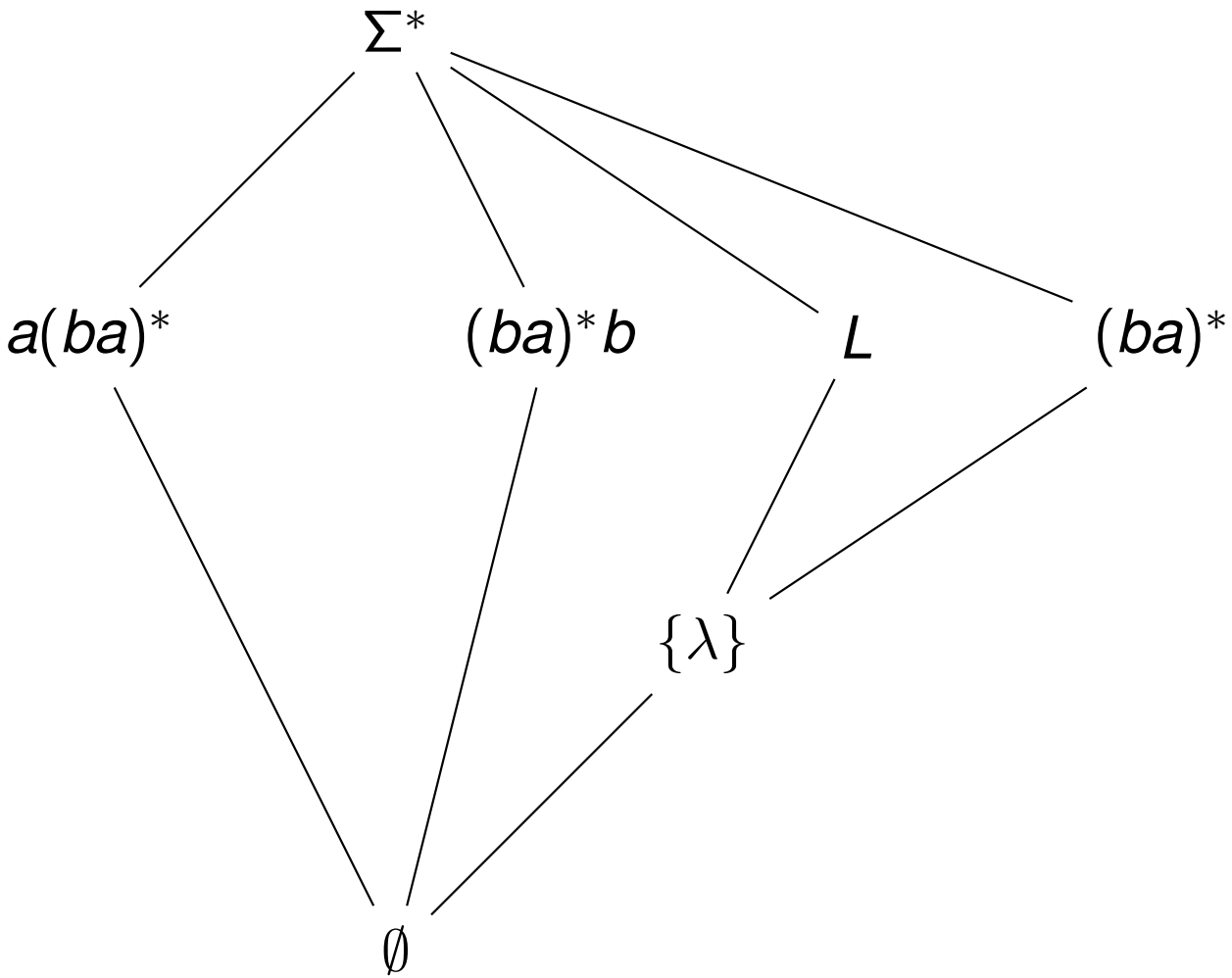
- We have a unique smallest CIS that recognises the language.
- This doesn't mean that we have a unique smallest CFG.
- But we do have a unique set of nonterminals that are now not arbitrary symbols but elements of a nontrivial algebraic structure.

The main point of this talk

Grammars without mergeable nonterminals will have nonterminals that correspond to closed sets of strings.

Simple Example

Suppose L is just regular: $(ab)^*$.



Linear Algebra 101

Dimensionality reduction

Linear map

$f : \mathbb{R}^n \to \mathbb{R}^k$, $k < n$

$f(\vec{x} + \vec{y}) = f(\vec{x}) + f(\vec{y})$ and $f(\alpha\vec{x}) = \alpha f(\vec{x})$

Linear Algebra 101

Dimensionality reduction

Linear map

$f : \mathbb{R}^n \to \mathbb{R}^k$, $k < n$

$f(\vec{x} + \vec{y}) = f(\vec{x}) + f(\vec{y})$ and $f(\alpha\vec{x}) = \alpha f(\vec{x})$

Linear maps are just vector space homomorphisms

- $\vec{x} \cong_f \vec{y}$ iff $f(\vec{x}) = f(\vec{y})$ iff $\vec{x} - \vec{y} \in \text{Ker}(f)$
- $\mathbb{R}^n / \cong_f$ is isomorphic to $f(\mathbb{R}^n)$ which is $\mathbb{R}^k$ if it is surjective.

But if k is too small, you lose information ...

Syntactic concept lattice

Alternative definition

$X \cong_L Y$ if $X^{\triangleright} = Y^{\triangleright}$

This is a CIS-congruence and $\mathfrak{B}(L)$ is just $\mathcal{P}(\Sigma^*)/\cong_L$

Syntactic concept lattice

Alternative definition

$X \cong_L Y$ if $X^{\triangleright} = Y^{\triangleright}$

This is a CIS-congruence and $\mathfrak{B}(L)$ is just $\mathcal{P}(\Sigma^*)/\cong_L$

Subspaces analogy

We are just modelling the smallest subspace that captures the phenomenon we are interested in.

But using as our algebra a CIS rather than a vector space.

Lambek, 1958

"We shall assign type n to all expressions which can occur in any context in which all proper names can occur."

- Let N be the set of proper names (not necessarily closed)
- $N^{\triangleright}$ is then the set of contexts that all proper nouns can occur in.
- $N^{\triangleright\triangleleft}$ is the set of all strings which can occur in any of $N^{\triangleright}$
- n is assigned to $N^{\triangleright\triangleleft}$ which is a concept.

Lambek, 1958

"We shall assign type n to all expressions which can occur in any context in which all proper names can occur."

- Let N be the set of proper names (not necessarily closed)
- $N^{\triangleright}$ is then the set of contexts that all proper nouns can occur in.
- $N^{\triangleright\triangleleft}$ is the set of all strings which can occur in any of $N^{\triangleright}$
- n is assigned to $N^{\triangleright\triangleleft}$ which is a concept.

- Types in Lambek grammars were intended to be closed sets of strings.
- SCLs form a sound and complete semantics for Lambek grammars (Wurm, 2012)

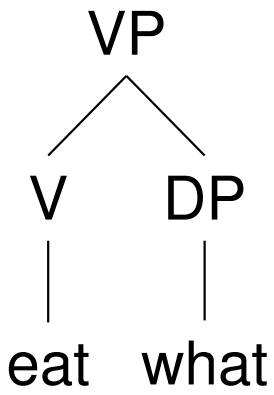
Outline

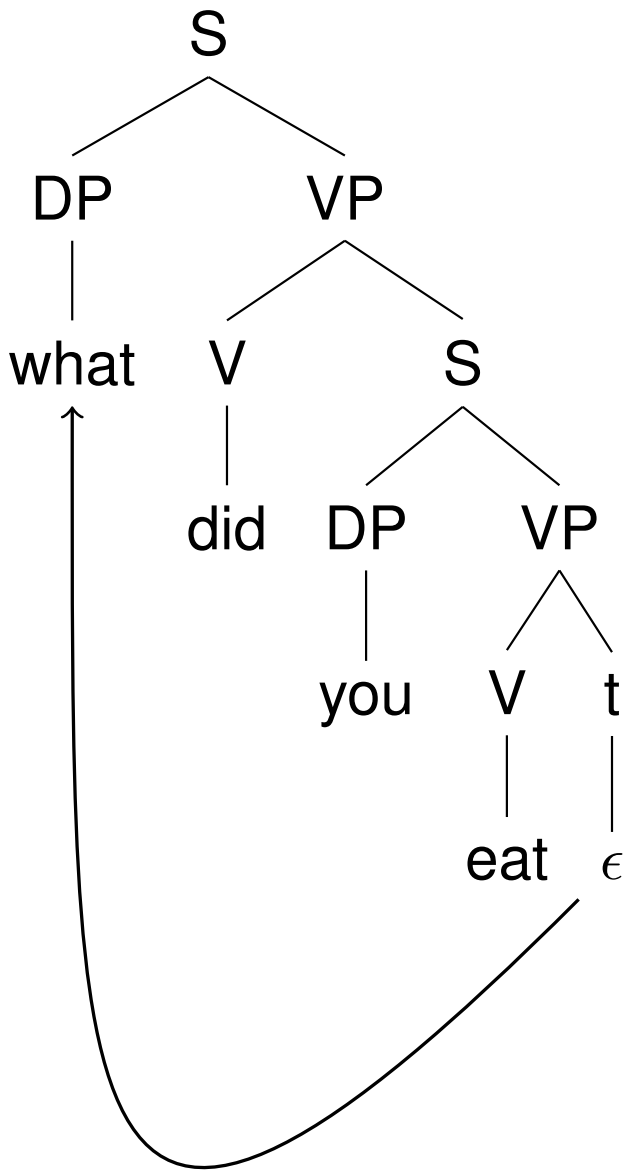
Multiple context free grammars

Mildly context-sensitive grammar formalisms

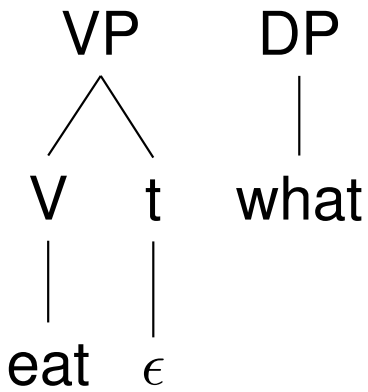
- Generalisation of context-free grammars where some nonterminals can generate tuples of strings.
- MCFGs of dimension 2 have nonterminals which generate pairs of strings: $N \stackrel{*}{\Rightarrow} \langle u, v \rangle$
- Essential for cross-serial dependencies in Dutch, Swiss German etc.
- Useful for modeling displacement in language with overt movement (English)

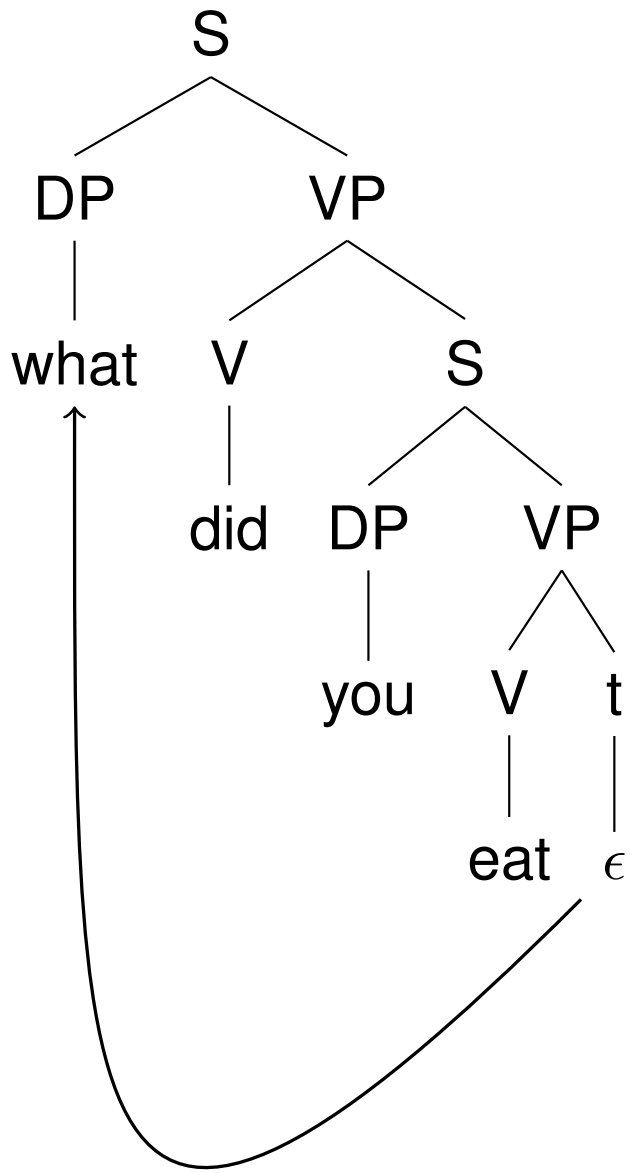
Movement





Movement





Syntactic concept lattice of order 2

2-contexts and 2-words

$l\Box m\Box r \odot \langle u, v \rangle = lumvr$

Given a set of pairs of strings X define

$$X^{\blacktriangleright} = \{l\Box m\Box r \mid l\Box m\Box r \odot X \subseteq L\}$$

Given a set of 2-contexts C we define

$$C^{\blacktriangleleft} = \{(u, v) \mid C \odot (u, v) \subseteq L\}$$

X is closed iff $X = X^{\blacktriangleright\blacktriangleleft}$.

Syntactic example

- Which book did Mary read?
- Which cake did Mary eat?

$(\square \text{ did Mary } \square \text{ ? }) \odot \langle \text{ which book, read} \rangle$
$(\square \text{ did Mary } \square \text{ ? }) \odot \langle \text{ which cake, eat} \rangle$

Minimality result

For a given MCFL_2, L, the minimal MCFG_2 will have nonterminals where:

- The nonterminals of dimension 1 will correspond to closed sets of strings.
- The nonterminals of dimension 2 will correspond to closed sets of pairs of strings.

Outline

Answers

What are the elements of $V = \{N, P, Q \dots\}$?

They are elements of the syntactic concept lattice.

What does S mean?

S is the concept/closed set of strings L.

What does a production like $N \to aPQ$ mean?

It refers to an inclusion relation $N \supseteq \mathcal{C}(a) \circ P \circ Q$

Syntactic structure and Strong learning

Hard question

We know that minimal (M)CFGs are *weakly* adequate for all (M)CFLs
Are the minimal grammars strongly adequate for natural language syntax?

Implications of using minimal grammars

- If two words *a* and *b* are distributionally identical then they must be syntactically identical.
- If a sentence *lar* has *n* distinct parses, then *lbr* must have *n* distinct parses of the same form.
- Very strong prediction that seems to be true, and is not made by any other theory.

Conclusion

- The syntactic concept lattice is a canonical object that provides a denotation for context free grammars; with a rich algebaric structure.
- The elements of it are maximal factorisations of the language into sets of contexts and sets of strings.
- We can compute (fragments of it) from knowledge of the strings in the language, which gives algorithms for weak and strong learning of CFGs.
- We can extend it to MCFGs;
- More generally we can view grammars algebraically; homomorphisms of the algebra give morphisms of the grammars.

Thanks

Thanks to . . .

Ryo Yoshinaka, Hans Leiss, Jason Eisner, . . .

References

“The syntactic concept lattice”, to appear (2013?), Journal of Logic and Computation.