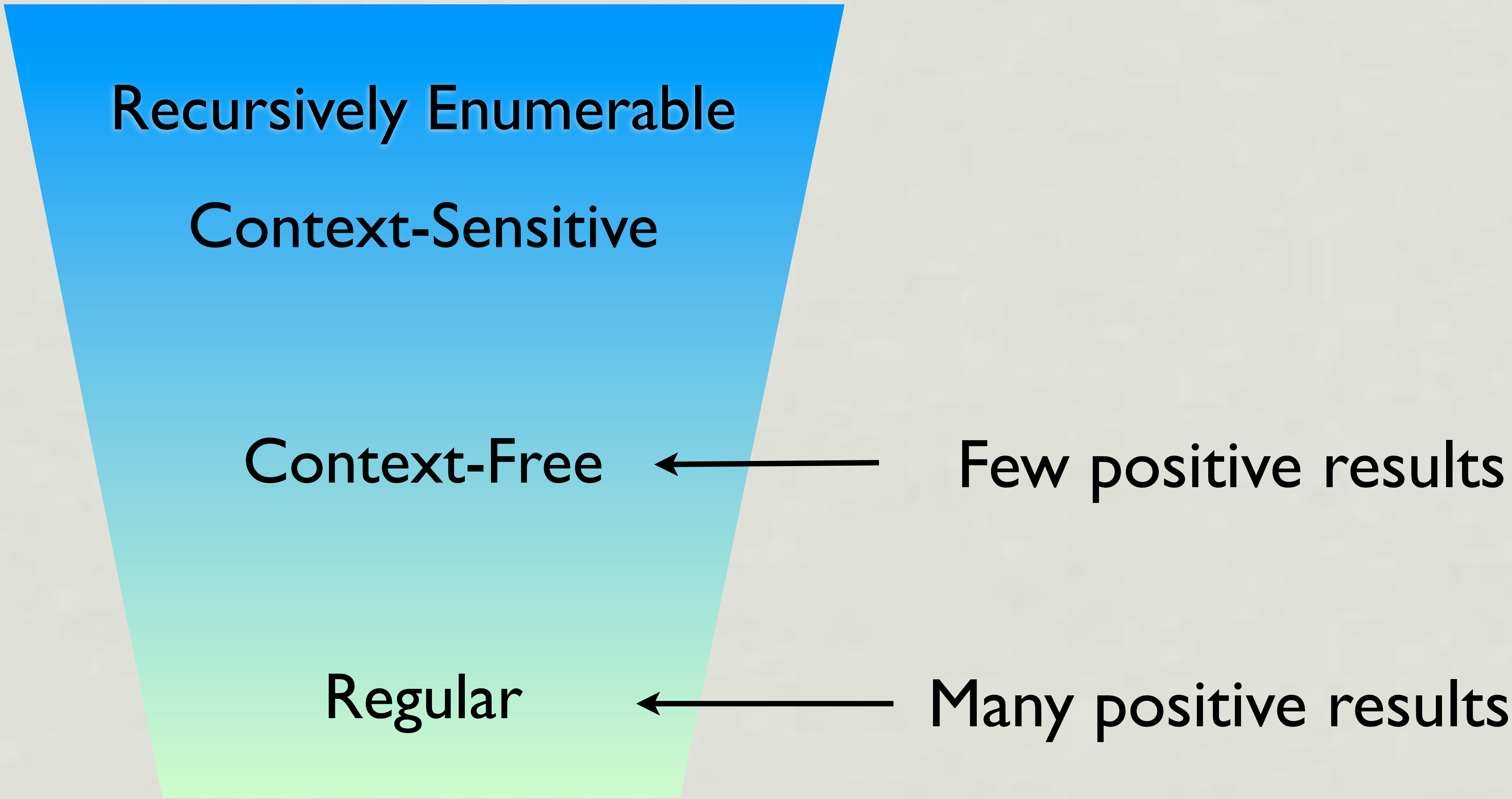
Distributional Learning of Multiple Context-Free Grammars and Related Formalisms

Ryo Yoshinaka
(ERATO Minato Discrete Structure Manipulation System Project,
Japan Science and Technology Agency)

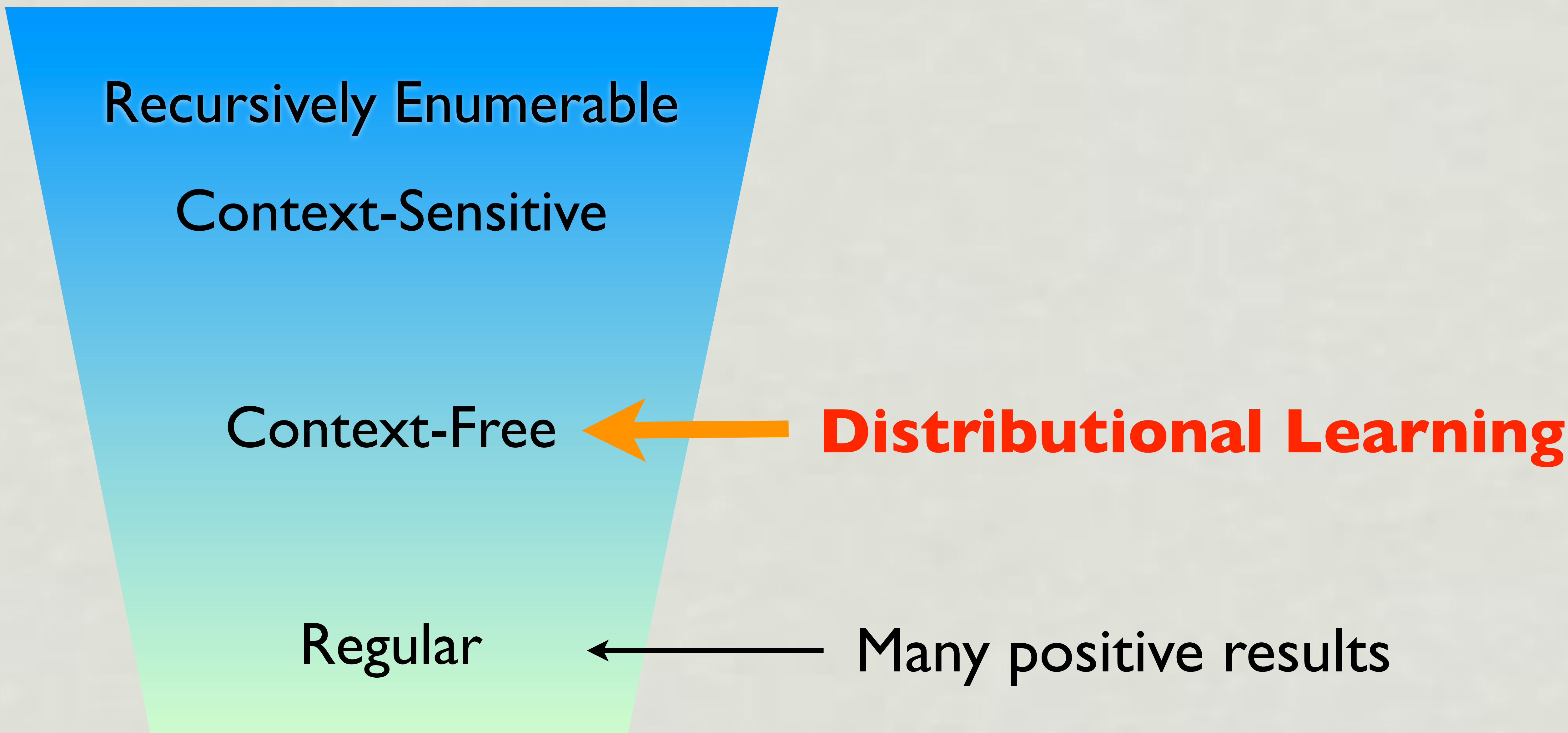
Distributional Learning of Multiple Context-Free Grammars and Related Formalisms Including MCFGs

Ryo Yoshinaka
(ERATO Minato Discrete Structure Manipulation System Project,
Japan Science and Technology Agency)

Chomsky Hierarchy & Learning



Chomsky Hierarchy & Learning



Distributional Learning

- Models and exploits the distribution of strings in contexts
- Syntactic category of a phrase = Contexts where it occurs

		contexts			
		John □ Mary	□ loves kids	A cat hits □	Everyone □
strings	John		◎	◎	
	Mary		◎	◎	
	she		◎		
	him			◎	
	loves	◎			
	loves it				◎
	runs				◎

John $\equiv$ Mary, him $\leqq$ Mary, loves it $\equiv$ runs, ...

Distributional Learning of CFLs

- Context-deterministic CFGs by queries
  (Shirakawa & Yokomori '93)
- (*k*,*l*-)Substitutable CFLs by positive data
  (Clark & Eyraud '05, Yoshinaka'08)
- Congruential CFGs by queries (Clark'10)
- (*p*-)Finite Kernel/Context-Property
  (Clark et al.'08, Clark '09, Clark '10, Yoshinaka'11)
- Probabilistic learning of Unambiguous
  (*k*,*l*-)NTS Languages (Clark'06, Luque'10)
- Inversion Transition Grammars (Clark'11)
- etc.

Distributional Learning of CFLs

- Context-deterministic CFGs by queries (Shirakawa & Yokomori '93)
- (*k*,*l*-)Substitutable CFLs by positive data (Clark & Eyraud '05, Yoshinaka'08)
- Congruential CFGs by queries (Clark'10)
- (*p*-)Finite Kernel/Context-Property (Clark et al.'08, Clark '09, Clark '10, Yoshinaka'11)
- Probabilistic learning of Unambiguous (*k*,*l*-)NTS Languages (Clark'06, Luque'10)
- Inversion Transition Grammars (Clark'11)
- etc.

Outline

0. Introduction

1. Learning Substitutable Context-Free Languages
   (Clark and Eyraud '05, '07)

2. Learning Context-Free Grammars
   with the p-Finite Kernel Property
   (Yoshinaka'11, Clark et al. '08,'09, Clark '10)

3. Extension to Multiple Context-Free Grammars

4. Extension to Related Formalisms

5. Conclusion

Learning of Substitutable Context-Free Languages from Positive Data

Terms

- ***Context***: pair of strings $(w_1, w_2) \in \Sigma^* \times \Sigma^*$

Terms

- ***Context***: pair of strings $(w_1,w_2) \in \Sigma^* \times \Sigma^*$
- $(w_1,w_2) \odot v = w_1\ v\ w_2$ for $v \in \Sigma^*$ and $(w_1,w_2) \in \Sigma^* \times \Sigma^*$

Terms

- ***Context***: pair of strings $(w_1,w_2) \in \Sigma^* \times \Sigma^*$
- $(w_1,w_2) \odot v = w_1\ v\ w_2$ for $v \in \Sigma^*$ and $(w_1,w_2) \in \Sigma^* \times \Sigma^*$
- $L/v = \{ w \in \Sigma^*\times\Sigma^* \mid w \odot v \in L \}$: **context set** of v w.r.t. L

Terms

- ***Context***: pair of strings $(w_1, w_2) \in \Sigma^* \times \Sigma^*$
- $(w_1, w_2) \odot v = w_1\, v\, w_2$ for $v \in \Sigma^*$ and $(w_1, w_2) \in \Sigma^* \times \Sigma^*$
- $L/v = \{ w \in \Sigma^* \times \Sigma^* \mid w \odot v \in L \}$: **context set** of v w.r.t. L
  - $(\varepsilon, \varepsilon) \in L/v$ iff $v \in L$

Terms

- **Context**: pair of strings $(w_1,w_2) \in \Sigma^* \times \Sigma^*$
- $(w_1,w_2) \odot v = w_1\ v\ w_2$ for $v \in \Sigma^*$ and $(w_1,w_2) \in \Sigma^* \times \Sigma^*$
- $L/v = \{ w \in \Sigma^* \times \Sigma^* \mid w \odot v \in L \}$: **context set** of v w.r.t. L
  - $(\varepsilon,\varepsilon) \in L/v$ iff $v \in L$
- u is **substitutable** for v in L
  $\Leftrightarrow$ $L/v \subseteq L/u$

  $\Leftrightarrow$ $w \odot v \in L$ implies $w \odot u \in L$ for every context w

Terms

- **Context**: pair of strings $(w_1,w_2) \in \Sigma^* \times \Sigma^*$
- $(w_1,w_2) \odot v = w_1\ v\ w_2$ for $v \in \Sigma^*$ and $(w_1,w_2) \in \Sigma^* \times \Sigma^*$
- $L/v = \{ w \in \Sigma^*{\times}\Sigma^* \mid w \odot v \in L \}$: **context set** of v w.r.t. L
  - $(\varepsilon,\varepsilon) \in L/v$ iff $v \in L$
- u is **substitutable** for v in L
  $\Leftrightarrow$ $L/v \subseteq L/u$
  $\Leftrightarrow$ $w \odot v \in L$ implies $w \odot u \in L$ for every context w
- u and v are **congruent** in L iff $L/v = L/u$

Terms

- **Context**: pair of strings $(w_1,w_2) \in \Sigma^* \times \Sigma^*$
- $(w_1,w_2) \odot v = w_1\ v\ w_2$ for $v \in \Sigma^*$ and $(w_1,w_2) \in \Sigma^* \times \Sigma^*$
- $L/v = \{\ w \in \Sigma^*\times\Sigma^* \mid w \odot v \in L\ \}$: **context set** of v w.r.t. L
  - $(\varepsilon,\varepsilon) \in L/v$ iff $v \in L$
- u is **substitutable** for v in L
  $\Leftrightarrow$ $L/v \subseteq L/u$
  $\Leftrightarrow$ $w \odot v \in L$ implies $w \odot u \in L$ for every context w
- u and v are **congruent** in L iff $L/v = L/u$
- ex. $L = \{\ a^nbc^n \mid n \geqq 0\ \}$.
  $L/abc = L/aabcc = \{\ (a^n,c^n) \mid n \geqq 0\ \}$

Easy Lemmas

- If u is substitutable for v, then $w \odot u$ is substitutable for $w \odot v$ for any context w

Easy Lemmas

- If u is substitutable for v, then $w \odot u$ is substitutable for $w \odot v$ for any context w
- $L/v_1 \subseteq L/u_1$ & $L/v_2 \subseteq L/u_2 \Rightarrow L/(v_1 v_2) \subseteq L/(u_1 u_2)$

Easy Lemmas

- If u is substitutable for v, then $w \odot u$ is substitutable for $w \odot v$ for any context w
- $L/v_1 \subseteq L/u_1$ & $L/v_2 \subseteq L/u_2 \Rightarrow L/(v_1v_2) \subseteq L/(u_1u_2)$

  $\because w_1v_1v_2w_2 \in L \Rightarrow w_1u_1v_2w_2 \in L \Rightarrow w_1u_1u_2w_2 \in L$

Easy Lemmas

- If u is substitutable for v, then $w \odot u$ is substitutable for $w \odot v$ for any context w
- $L/v_1 \subseteq L/u_1$ & $L/v_2 \subseteq L/u_2 \Rightarrow L/(v_1v_2) \subseteq L/(u_1u_2)$

  $\because w_1v_1v_2w_2 \in L \Rightarrow w_1u_1v_2w_2 \in L \Rightarrow w_1u_1u_2w_2 \in L$
- $L/v_1 = L/u_1$ & $L/v_2 = L/u_2 \Rightarrow L/(v_1v_2) = L/(u_1u_2)$

Substitutable CFLs

- Clark and Eyraud ('05,'07)
- L is **substitutable** iff
  $\forall v_1, v_2 \in \Sigma^+, \quad L/v_1 \cap L/v_2 \neq \varnothing$ implies $L/v_1 = L/v_2$

Substitutable CFLs

- Clark and Eyraud ('05,'07)
- L is **substitutable** iff
  $\forall v_1, v_2 \in \Sigma^+, \quad L/v_1 \cap L/v_2 \neq \varnothing$ implies $L/v_1 = L/v_2$
- Positive data:

Substitutable CFLs

- Clark and Eyraud ('05,'07)
- *L* is **substitutable** iff
  $\forall v_1, v_2 \in \Sigma^+, \quad L/v_1 \cap L/v_2 \neq \varnothing$ implies $L/v_1 = L/v_2$
- Positive data:
  - A man gave John chocolate.
  - A man gave a little girl chocolate.

Substitutable CFLs

- Clark and Eyraud ('05,'07)
- *L* is **substitutable** iff
  $\forall v_1, v_2 \in \Sigma^+, \quad L/v_1 \cap L/v_2 \neq \emptyset$ implies $L/v_1 = L/v_2$
- Positive data:
  - A man gave John chocolate.
  - A man gave a little girl chocolate.
  - They like John.

Substitutable CFLs

- Clark and Eyraud ('05,'07)
- L is **substitutable** iff
  $\forall v_1, v_2 \in \Sigma^+, \quad L/v_1 \cap L/v_2 \neq \varnothing$ implies $L/v_1 = L/v_2$
- Positive data:
    - A man gave John chocolate.
    - A man gave a little girl chocolate.
    - They like John.
- Generalization: They like a little girl.

Substitutable CFLs

- Clark and Eyraud ('05,'07)
- L is **substitutable** iff
  $\forall v_1, v_2 \in \Sigma^+$, $L/v_1 \cap L/v_2 \neq \varnothing$ implies $L/v_1 = L/v_2$
- Positive data:
  - A man gave John chocolate.
  - A man gave a little girl chocolate.
  - They like John.
- Generalization: They like a little girl.
- ex. $L = \{ a^n b c^n \mid n \geqq 0 \}$ is substitutable.

Identification in the Limit from Positive Data

- Gold (1967)

Identification in the Limit from Positive Data

- Gold (1967)

Learning Target

L_0

Identification in the Limit from Positive Data

- Gold (1967)
- Learner
  - gets a positive example $w_1 \quad w_2 \quad w_3 \quad w_4 \quad \ldots$
  - updates the conjecture $G_1 \quad G_2 \quad G_3 \quad G_4 \quad \ldots$
  - $L_0 = \{ w_1, w_2, w_3, \ldots \}$

Learning Target

L_0

Identification in the Limit from Positive Data

- Gold (1967)
- Learner
  - gets a positive example w_1 w_2 w_3 w_4 ...
  - updates the conjecture G_1 G_2 G_3 G_4 ...
  - $L_0 = \{ w_1, w_2, w_3, \ldots \}$

Learning Target L_0

- Identification in the Limit:
  - convergence to a grammar for the target
    $G_n = G_{n+1} = G_{n+2} \ldots$ and $L(G_n) = L_0$
- Learner should uniformly learn a rich class of languages

Clark & Eyraud's Algorithm

```
let G := vacuous grammar;
For n = 1,2,3,...
  let D := {w1 , w2 , ... , wn };
  If D ⊈ L(G)
    then update G by D;
  End if
  output G
End for
```

Learner's Conjecture

- $D = \{ u_1, ..., u_n \}$: positive data

Learner's Conjecture

- $D = \{ u_1, ..., u_n \}$: positive data
- G_D : conjecture

Learner's Conjecture

- $D = \{ u_1, \ldots, u_n \}$: positive data
- G_D : conjecture
  - $N = \mathrm{Sub}(D)$: all non-empty substrings from D
    $= \{ [\![v]\!] \mid \exists w, w \odot v \in D, v \in \Sigma^+ \}$

Learner's Conjecture

- $D = \{ u_1, ..., u_n \}$: positive data
- G_D : conjecture
  - $N = \mathrm{Sub}(D)$: all non-empty substrings from D
    $= \{ [\![v]\!] \mid \exists w, w \odot v \in D, v \in \Sigma^+ \}$

★ $[\![v]\!] \Rightarrow u$ for $L_0/v = L_0/u$,

Learner's Conjecture

- $D = \{ u_1, ..., u_n \}$: positive data
- G_D : conjecture
  - N = Sub(D) : all non-empty substrings from D
    $= \{ [\![v]\!] \mid \exists w,\ w \odot v \in D, v \in \Sigma^+ \}$

★ $[\![v]\!] \Rightarrow u$ for $L_0/v = L_0/u$,

  - Initial Symbols: $\{ [\![v]\!] \in N \mid v \in D \}$

Learner's Conjecture

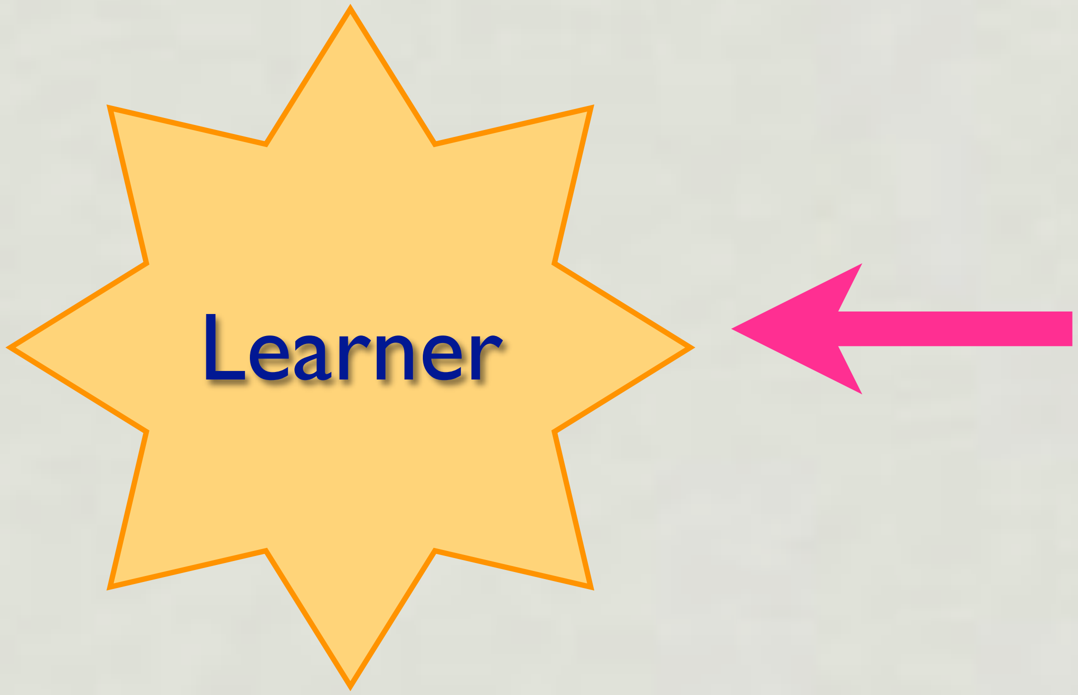
- $D = \{ u_1, ..., u_n \}$: positive data
- G_D : conjecture
  - $N = \mathrm{Sub}(D)$: all non-empty substrings from D
    $= \{ [\![v]\!] \mid \exists w, w \odot v \in D, v \in \Sigma^+ \}$

★ $[\![v]\!] \Rightarrow u$ for $L_0/v = L_0/u$,

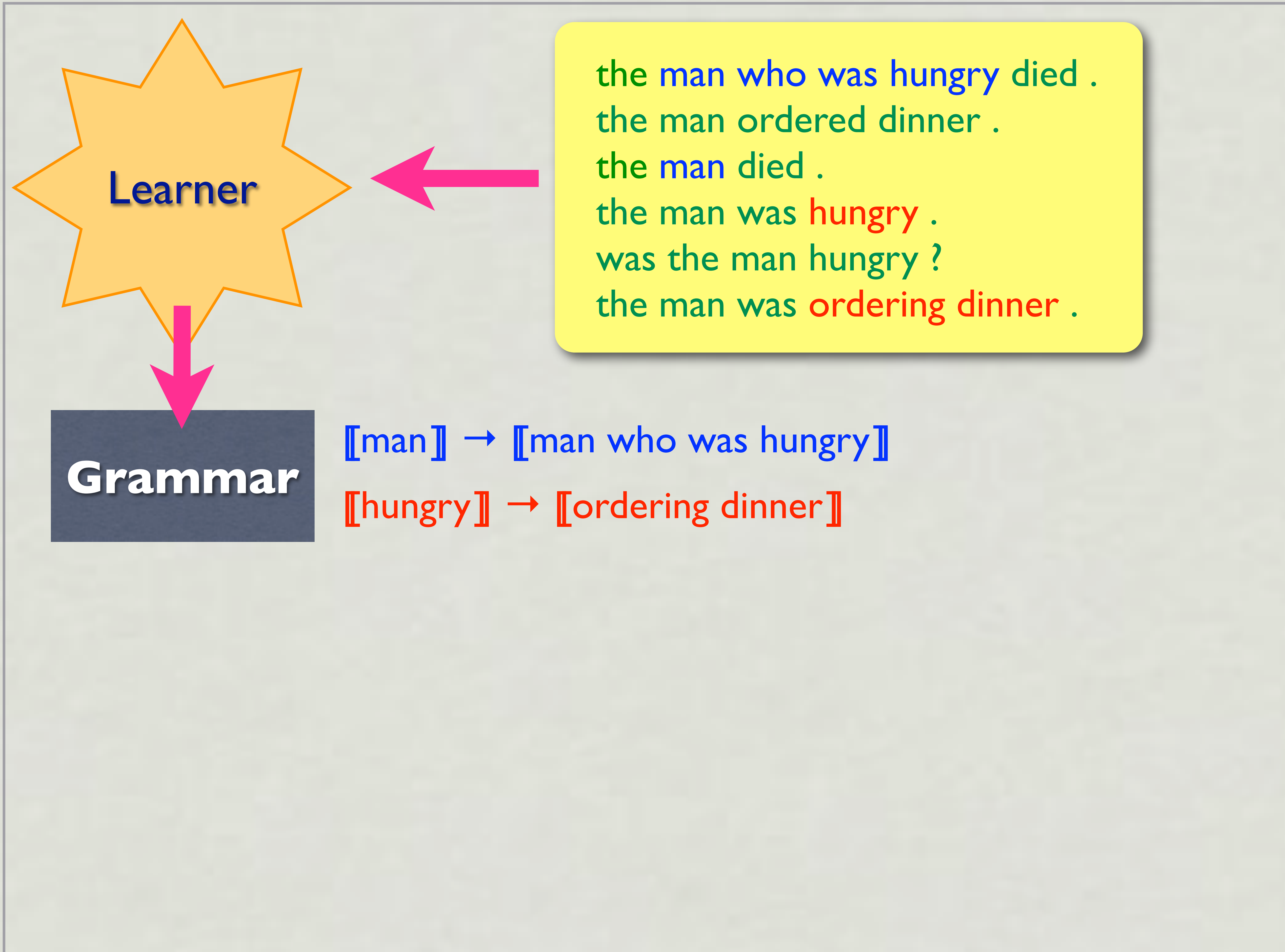
  - Initial Symbols: $\{ [\![v]\!] \in N \mid v \in D \}$
  - Rules
    - Type I: $[\![v_1 v_2]\!] \to [\![v_1]\!][\![v_2]\!]$ for all $[\![v_1 v_2]\!] \in N$
    - Type II: $[\![a]\!] \to a$ for all $a \in \Sigma$

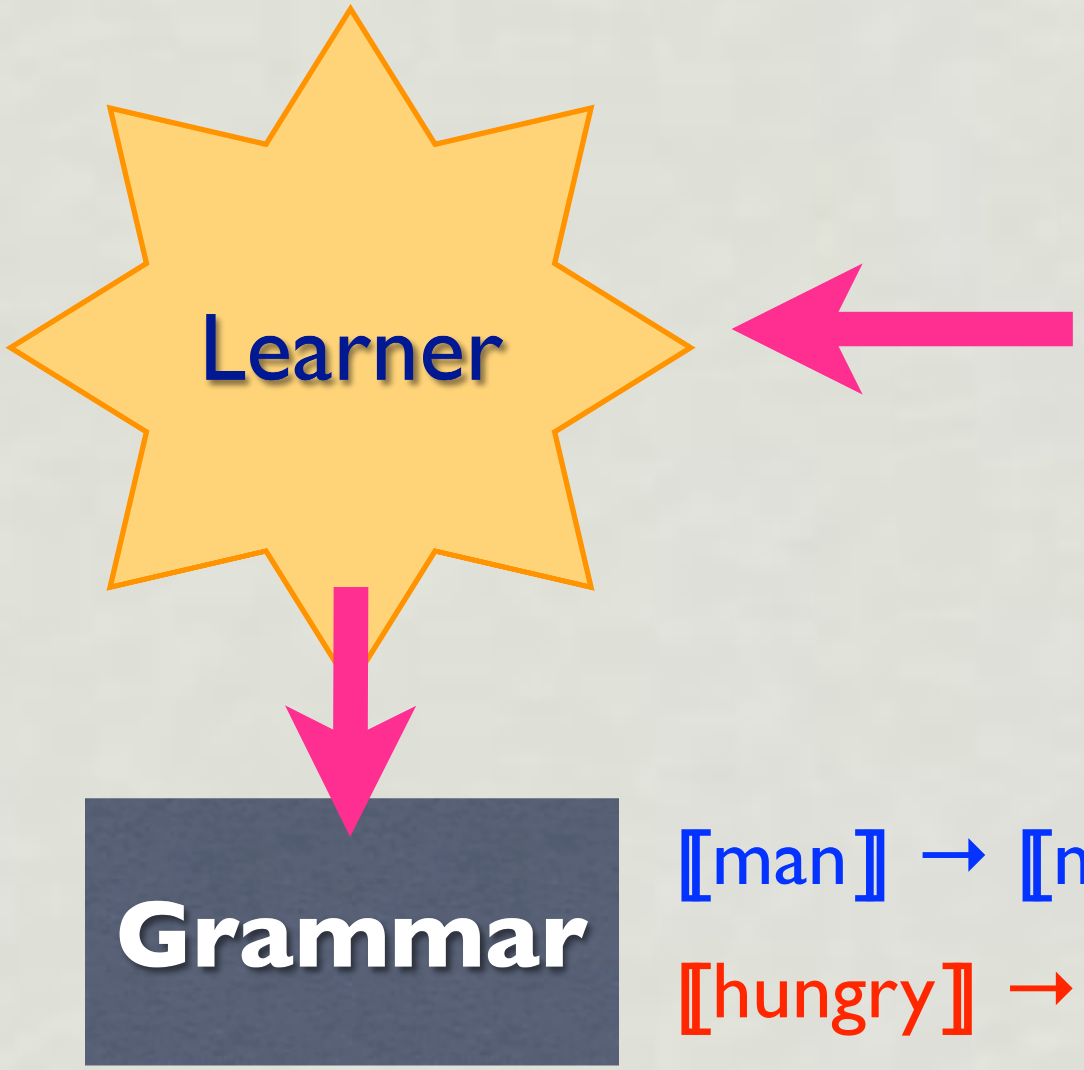
Learner's Conjecture

- $D = \{ u_1, ..., u_n \}$: positive data
- G_D : conjecture
  - $N = \mathrm{Sub}(D)$: all non-empty substrings from D
    $= \{ [\![v]\!] \mid \exists w, w \odot v \in D, v \in \Sigma^+ \}$
  - ★ $[\![v]\!] \Rightarrow u$ for $L_0/v = L_0/u$,
  - Initial Symbols: $\{ [\![v]\!] \in N \mid v \in D \}$
  - Rules
    - Type I: $[\![v_1 v_2]\!] \to [\![v_1]\!][\![v_2]\!]$ for all $[\![v_1 v_2]\!] \in N$
    - Type II: $[\![a]\!] \to a$ for all $a \in \Sigma$
    - Type III: $[\![v]\!] \to [\![u]\!]$ if $\exists\, w$ s.t. $w \odot v, w \odot u \in D$ $(\subseteq L_0)$



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


Learner
the man who was hungry died .
the man ordered dinner .
the man died .
the man was hungry .
was the man hungry ?
the man was ordering dinner .
Grammar
⟦man⟧ → ⟦man who was hungry⟧
⟦hungry⟧ → ⟦ordering dinner⟧



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

⟦man⟧ → ⟦man who was hungry⟧

⟦hungry⟧ → ⟦ordering dinner⟧

was the man who was hungry ordering dinner ?

Learner

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

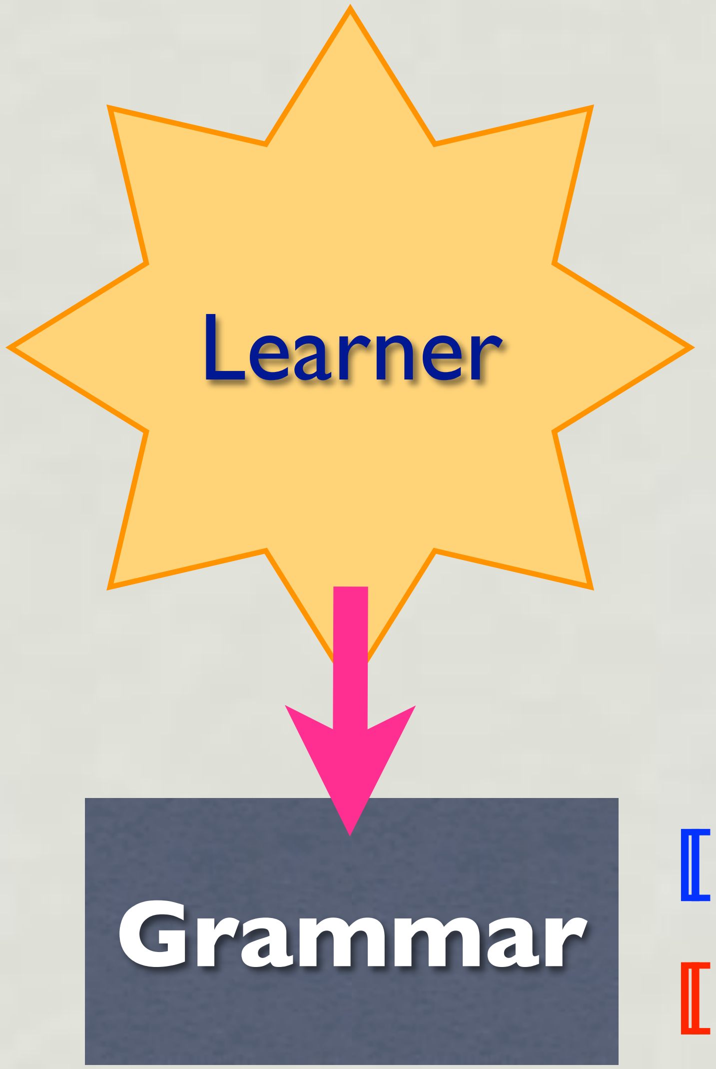
Grammar

⟦man⟧ → ⟦man who was hungry⟧

⟦hungry⟧ → ⟦ordering dinner⟧

⟦was the man hungry ?⟧ ⇒ ⟦was the man⟧ ⟦hungry ?⟧

was the man who was hungry ordering dinner ?



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

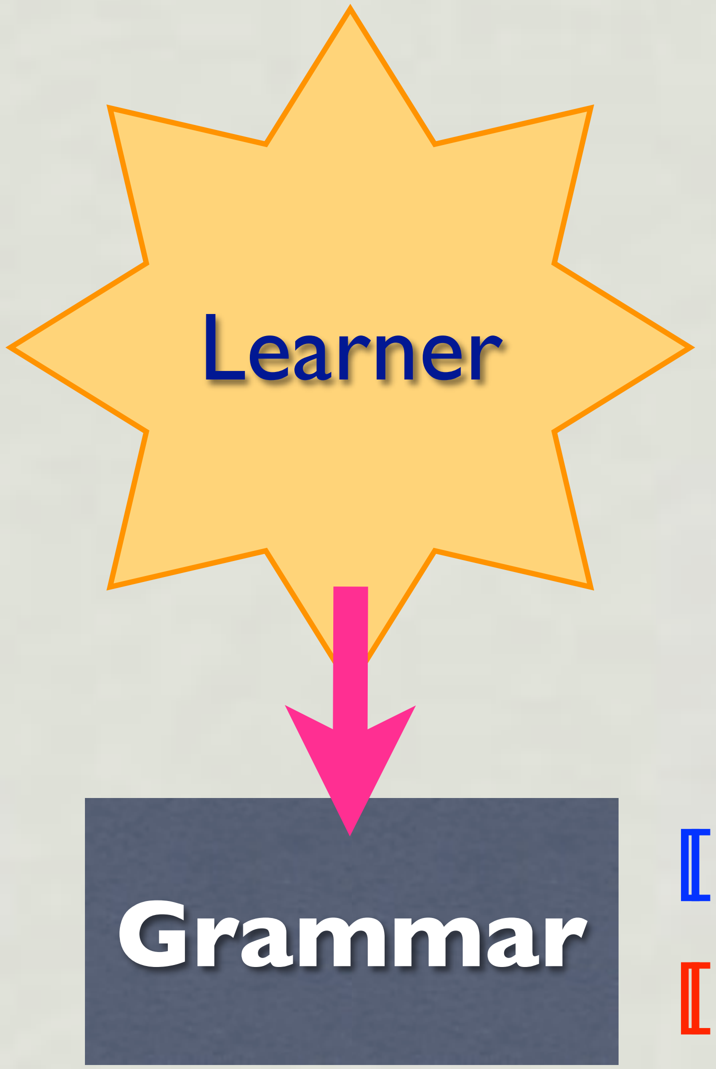
⟦man⟧ → ⟦man who was hungry⟧

⟦hungry⟧ → ⟦ordering dinner⟧

⟦was the man hungry ?⟧ ⇒ ⟦was the man⟧ ⟦hungry ?⟧

⇒ ⟦was the man⟧ ⟦hungry⟧ ⟦?⟧ ⇒ ⟦was⟧ ⟦the⟧ ⟦man⟧ ⟦hungry⟧ ⟦?⟧

was the man who was hungry ordering dinner ?



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

⟦man⟧ → ⟦man who was hungry⟧

⟦hungry⟧ → ⟦ordering dinner⟧

⟦was the man hungry ?⟧ ⇒ ⟦was the man⟧⟦hungry ?⟧

⇒ ⟦was the man⟧⟦hungry⟧⟦?⟧ ⇒ ⟦was⟧⟦the⟧⟦man⟧⟦hungry⟧⟦?⟧

⇒ ⟦was⟧⟦the⟧⟦man⟧⟦ordering dinner⟧⟦?⟧

was the man who was hungry ordering dinner ?

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

Learner

Grammar

⟦man⟧ → ⟦man who was hungry⟧

⟦hungry⟧ → ⟦ordering dinner⟧

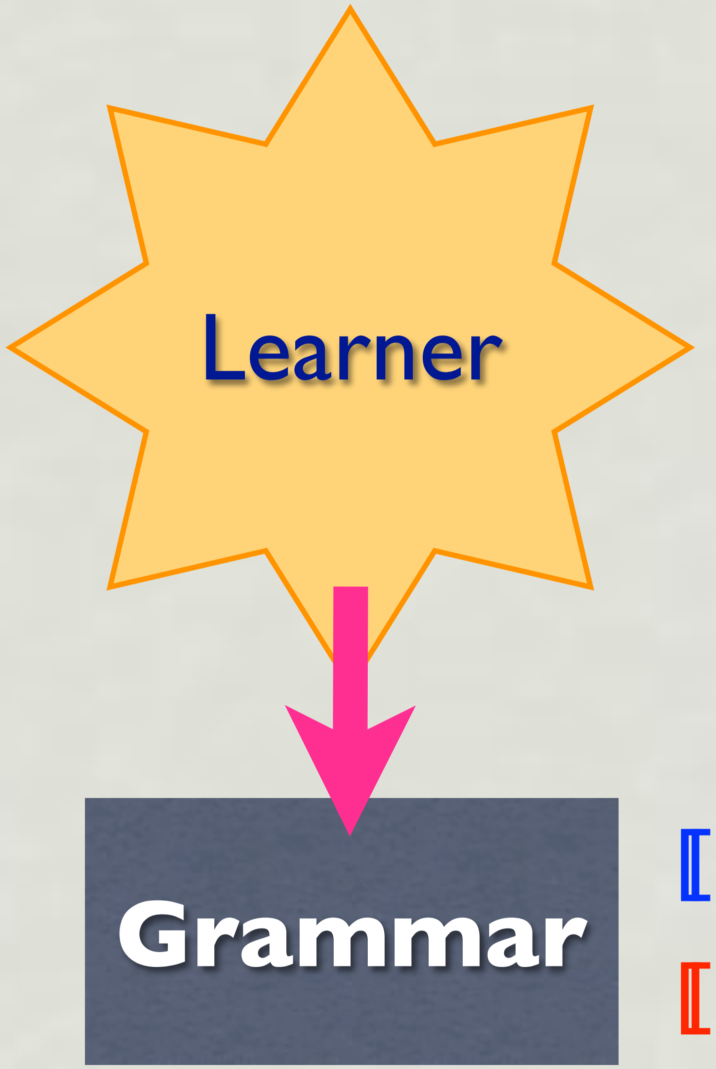
⟦was the man hungry ?⟧ ⇒ ⟦was the man⟧⟦hungry ?⟧

⇒ ⟦was the man⟧⟦hungry⟧⟦?⟧ ⇒ ⟦was⟧⟦the⟧⟦man⟧⟦hungry⟧⟦?⟧

⇒ ⟦was⟧⟦the⟧⟦man⟧⟦ordering dinner⟧⟦?⟧

⇒ ⟦was⟧⟦the⟧⟦man who was hungry⟧⟦ordering dinner⟧⟦?⟧

was the man who was hungry ordering dinner ?



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

⟦man⟧ → ⟦man who was hungry⟧

⟦hungry⟧ → ⟦ordering dinner⟧

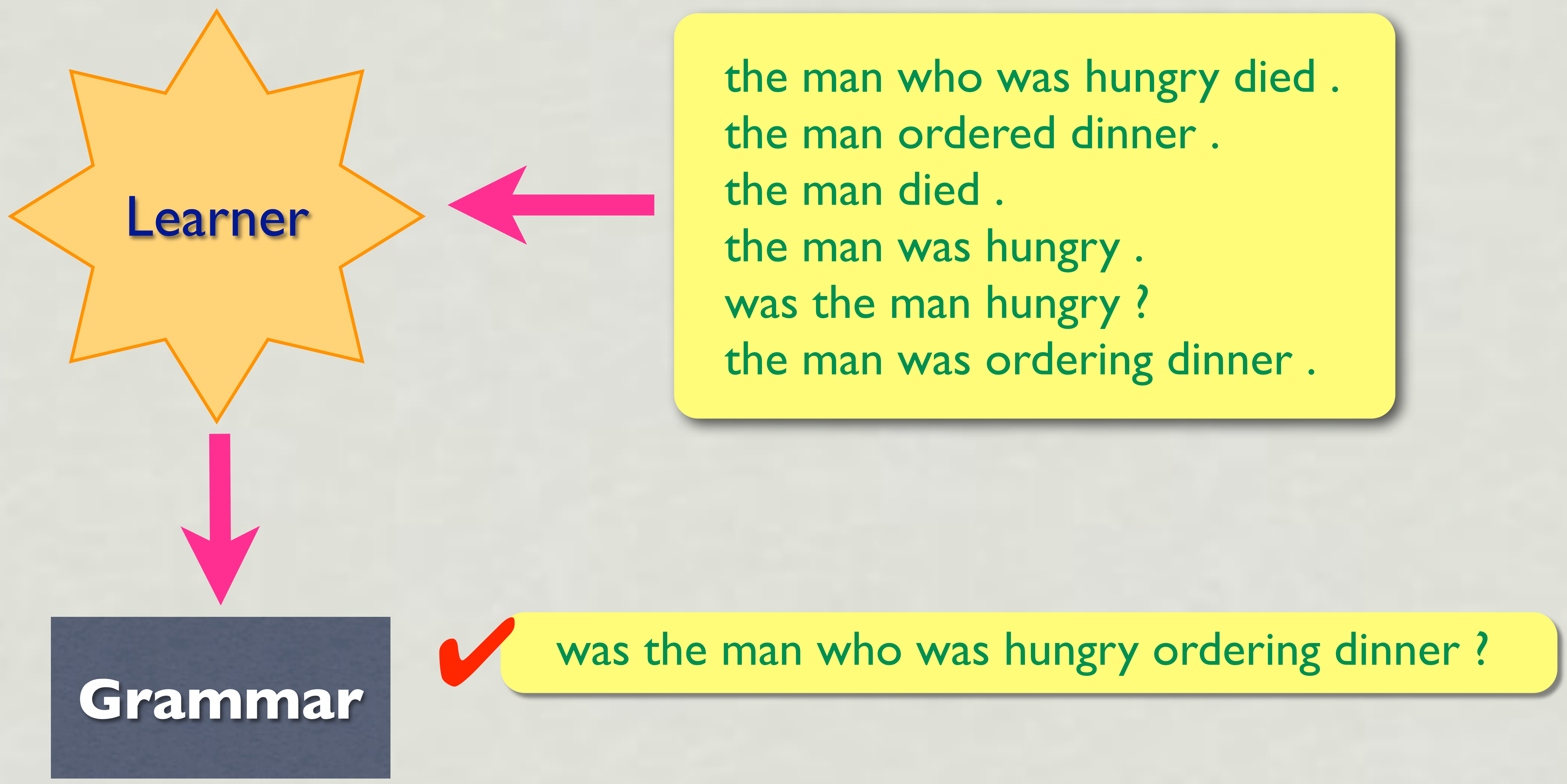
⟦was the man hungry ?⟧ ⇒ ⟦was the man⟧⟦hungry ?⟧

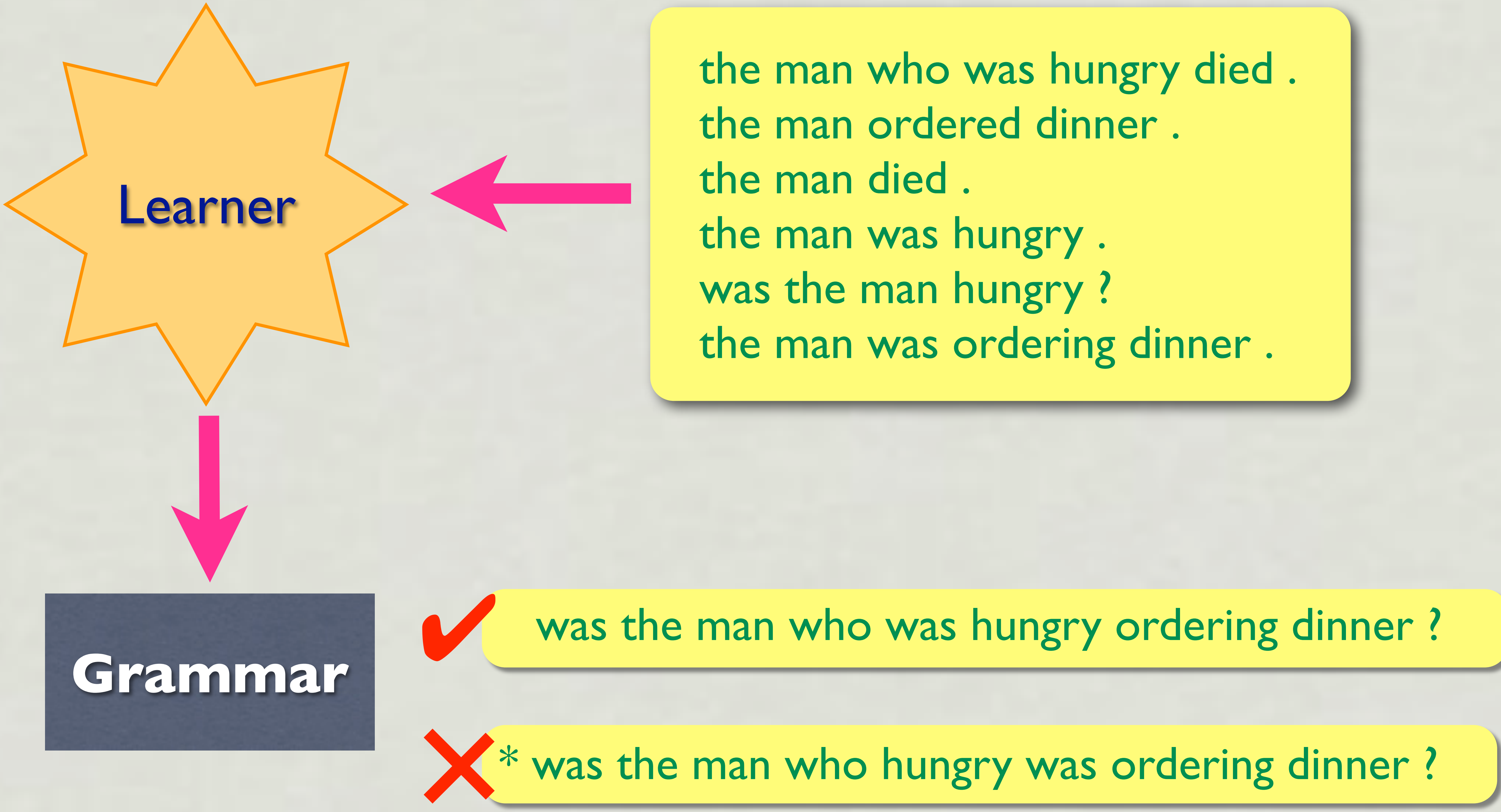
⇒ ⟦was the man⟧⟦hungry⟧⟦?⟧ ⇒ ⟦was⟧⟦the⟧⟦man⟧⟦hungry⟧⟦?⟧

⇒ ⟦was⟧⟦the⟧⟦man⟧⟦ordering dinner⟧⟦?⟧

⇒ ⟦was⟧⟦the⟧⟦man who was hungry⟧⟦ordering dinner⟧⟦?⟧

⇒ was the man who was hungry ordering dinner ?


Learner
the man who was hungry died .
the man ordered dinner .
the man died .
the man was hungry .
was the man hungry ?
the man was ordering dinner .
Grammar
was the man who was hungry ordering dinner ?


Learner
the man who was hungry died .
the man ordered dinner .
the man died .
the man was hungry .
was the man hungry ?
the man was ordering dinner .
Grammar
was the man who was hungry ordering dinner ?
* was the man who hungry was ordering dinner ?

Soundness

- G_D : conjecture, L_0 : target language
- Proposition: $L(G_D) \subseteq L_0$

Soundness

- G_D : conjecture, L_0 : target language
- Proposition: $L(G_D) \subseteq L_0$
  - homomorphism h: $h(a) = a$ *for* $a \in \Sigma$, $h([\![v]\!]) = v$ *for* $v \in \Sigma^+$,

Soundness

- G_D : conjecture, L_0 : target language
- Proposition: $L(G_D) \subseteq L_0$
  - homomorphism h: $h(a) = a$ *for* $a \in \Sigma$, $h(\llbracket v \rrbracket) = v$ *for* $v \in \Sigma^+$,
  - If $\alpha \Rightarrow \beta$ in G_D, then $L_0/h(\alpha) = L_0/h(\beta)$

Soundness

- G_D : conjecture, L_0 : target language
- Proposition: $L(G_D) \subseteq L_0$
  - homomorphism h: $h(a) = a$ for $a \in \Sigma$, $h([\![v]\!]) = v$ for $v \in \Sigma^+$,
  - If $\alpha \Rightarrow \beta$ in G_D, then $L_0/h(\alpha) = L_0/h(\beta)$
    - Type I: $[\![v_1 v_2]\!] \to [\![v_1]\!][\![v_2]\!]$ for $[\![v_1 v_2]\!], [\![v_1]\!], [\![v_2]\!] \in N$

Soundness

- G_D : conjecture, L_0 : target language
- Proposition: $L(G_D) \subseteq L_0$
  - homomorphism h: $h(a) = a$ for $a \in \Sigma$, $h([\![v]\!]) = v$ for $v \in \Sigma^+$,
  - If $\alpha \Rightarrow \beta$ in G_D, then $L_0/h(\alpha) = L_0/h(\beta)$
    - Type I: $[\![v_1 v_2]\!] \to [\![v_1]\!][\![v_2]\!]$ for $[\![v_1 v_2]\!], [\![v_1]\!], [\![v_2]\!] \in N$
    - Type II: $[\![a]\!] \to a$ for all $a \in \Sigma$

Soundness

- G_D : conjecture, L_0 : target language
- Proposition: $L(G_D) \subseteq L_0$
  - homomorphism h: $h(a) = a$ for $a \in \Sigma$, $h([\![v]\!]) = v$ for $v \in \Sigma^+$,
  - If $\alpha \Rightarrow \beta$ in G_D, then $L_0/h(\alpha) = L_0/h(\beta)$
    - Type I: $[\![v_1 v_2]\!] \to [\![v_1]\!][\![v_2]\!]$ for $[\![v_1 v_2]\!], [\![v_1]\!], [\![v_2]\!] \in N$
    - Type II: $[\![a]\!] \to a$ for all $a \in \Sigma$
    - Type III: $[\![v]\!] \to [\![u]\!]$ if $\exists\, w$ s.t. $w \odot v, w \odot u \in D$ $(\subseteq L_0)$

Soundness

- G_D : conjecture, L_0 : target language
- Proposition: $L(G_D) \subseteq L_0$
  - homomorphism h: $h(a) = a$ for $a \in \Sigma$, $h([\![v]\!]) = v$ for $v \in \Sigma^+$,
  - If $\alpha \Rightarrow \beta$ in G_D, then $L_0/h(\alpha) = L_0/h(\beta)$
    - Type I: $[\![v_1 v_2]\!] \to [\![v_1]\!][\![v_2]\!]$ for $[\![v$ …
    - Type II: $[\![a]\!] \to a$ for all $a \in \Sigma$
    - Type III: $[\![v]\!] \to [\![u]\!]$ if $\exists\, w$ s.t. $w \odot v, w \odot u \in D$ ($\subseteq L_0$)

$L/v = L/u$

$\Rightarrow L/(x \odot v) = L/(x \odot u)$

Soundness

- G_D : conjecture, L_0 : target language
- Proposition: $L(G_D) \subseteq L_0$
  - homomorphism h: $h(a) = a$ for $a \in \Sigma$, $h([\![v]\!]) = v$ for $v \in \Sigma^+$,
  - If $\alpha \Rightarrow \beta$ in G_D, then $L_0/h(\alpha) = L_0/h(\beta)$
    - Type I: $[\![v_1 v_2]\!] \to [\![v_1]\!][\![v_2]\!]$ for $[\![v$
    - Type II: $[\![a]\!] \to a$ for all $a \in \Sigma$
    - Type III: $[\![v]\!] \to [\![u]\!]$ if $\exists\, w$ s.t. $w \odot v, w \odot u \in D$ $(\subseteq L_0)$
  - For $\alpha = [\![v]\!] \in I$ and $\beta \in \Sigma^+$, v and β are congruent

$L/v = L/u$

$\Rightarrow L/(x \odot v) = L/(x \odot u)$

Completeness

- G_D : conjecture, G_0 : grammar in CNF s.t. $L_0 = L(G_0)$

Completeness

- G_D : conjecture, G_0 : grammar in CNF s.t. $L_0 = L(G_0)$
- v_α : shortest in $L(G_0,\alpha) = \{ v \mid \alpha \Rightarrow v \text{ in } G_0 \}$

Completeness

- G_D : conjecture, G_0 : grammar in CNF s.t. L_0 = L(G_0)
- v_α : shortest in L(G_0,α) = { v | $\alpha \Rightarrow v$ *in* G_0 }
- w_A : shortest context for a nonterminal A s.t. $S \Rightarrow w_A \odot A$

Completeness

- G_D : conjecture, G_0 : grammar in CNF s.t. $L_0 = L(G_0)$
- v_α : shortest in $L(G_0,\alpha) = \{ v \mid \alpha \Rightarrow v \text{ in } G_0 \}$
- w_A : shortest context for a nonterminal A s.t. $S \Rightarrow w_A \odot A$
- If $\{ w_A \odot v_\alpha \mid A \to \alpha \text{ in } G_0 \} \subseteq D$, then $L_0 \subseteq L(G_D)$.

Completeness

- G_D : conjecture, G_0 : grammar in CNF s.t. L_0 = L(G_0)
- v_α : shortest in L(G_0,α) = { v | $\alpha \Rightarrow v$ *in* G_0 }
- w_A : shortest context for a nonterminal A s.t. $S \Rightarrow w_A \odot A$
- If { $w_A \odot v_\alpha$ | $A \rightarrow \alpha$ in G_0 } $\subseteq D$, then $L_0 \subseteq$ L(G_D).
  - For $A \rightarrow BC$ in G_0 , $w_A \odot v_A$, $w_A \odot v_B v_C \in D$. Thus

Completeness

- G_D : conjecture, G_0 : grammar in CNF s.t. $L_0 = L(G_0)$
- v_α : shortest in $L(G_0,\alpha) = \{ v \mid \alpha \Rightarrow v \text{ in } G_0 \}$
- w_A : shortest context for a nonterminal A s.t. $S \Rightarrow w_A \odot A$
- If $\{ w_A \odot v_\alpha \mid A \to \alpha \text{ in } G_0 \} \subseteq D$, then $L_0 \subseteq L(G_D)$.
  - For $A \to BC$ in G_0 , $w_A \odot v_A$, $w_A \odot v_B v_C \in D$. Thus
  - $[\![v_A]\!] \Rightarrow [\![v_B v_C]\!] \Rightarrow [\![v_B]\!][\![v_C]\!]$ in G_D

Completeness

- G_D : conjecture, G_0 : grammar in CNF s.t. L_0 = L(G_0)
- v_α : shortest in L(G_0,α) = { v | $\alpha \Rightarrow v$ *in* G_0 }
- w_A : shortest context for a nonterminal A s.t. $S \Rightarrow w_A \odot A$
- If { $w_A \odot v_\alpha$ | $A \rightarrow \alpha$ in G_0 } $\subseteq D$, then $L_0 \subseteq$ L(G_D).
  - For $A \rightarrow BC$ in G_0 , $w_A \odot v_A$, $w_A \odot v_B v_C \in D$. Thus
  - $[\![v_A]\!] \Rightarrow [\![v_B v_C]\!] \Rightarrow [\![v_B]\!][\![v_C]\!]$ in G_D
- $[\![v_S]\!]$ is an initial symbol for $v_S \in D$

Theorem

- Clark and Eyraud's algorithm identifies every Substitutable CFL in the limit from positive data
- Polynomial-time update
- Polynomially many examples are enough for convergence w.r.t. the size of the grammar to be learnt

Substitutable Languages

- $\{ a^n b c^n \mid n \geqq 0 \}$ is substitutable
- $L = \{ a^n c^n \mid n \geqq 0 \}$ is not,
  because $(\varepsilon, c) \in L/a \cap L/aac$ but $(\varepsilon, acc) \in L/a - L/aac$.
- $a, ab \in L'$ implies $ab^* \subseteq L'$

Learning of CFGs with the *p*-Finite Kernel Property from Positive Data & Membership Queries

- For $L \subseteq \Sigma^*$ and $V \subseteq \Sigma^*$, the **context set** of V is $L/V = \bigcap_{v \in V} L/v = \{ w \in \Sigma^* \times \Sigma^* \mid w \odot V \subseteq L \}$
  - $U \subseteq V$ implies $L/U \supseteq L/V$

- For $L \subseteq \Sigma^*$ and $V \subseteq \Sigma^*$, the **context set** of V is $L/V = \bigcap_{v \in V} L/v = \{ w \in \Sigma^* \times \Sigma^* \mid w \odot V \subseteq L \}$
  - $U \subseteq V$ implies $L/U \supseteq L/V$

Example: $L = \{ a^i b^j c^k \mid i=j \text{ or } j=k \}$

L		contexts $\in \Sigma^* \times \Sigma^*$					
		$(\varepsilon,\varepsilon)$	(a,ε)	(ε,c)	(a,c)	(a,bc)	(ab,c)
strings $\in \Sigma^*$	ε	✓	✓	✓	✓	✓	✓
	a	✓	✓			✓	
	b		✓	✓	✓		
	c	✓		✓			✓
	ab	✓		✓	✓	✓	
	bc	✓	✓		✓		✓
	abc	✓	✓	✓			
	$aabb$	✓		✓		✓	
	$bbcc$	✓	✓				✓

- For $L \subseteq \Sigma^*$ and $V \subseteq \Sigma^*$, the **context set** of V is
  $L/V = \bigcap_{v \in V} L/v = \{ w \in \Sigma^* \times \Sigma^* \mid w \odot V \subseteq L \}$
  - $U \subseteq V$ implies $L/U \supseteq L/V$

Example: $L = \{ a^i b^j c^k \mid i{=}j \text{ or } j{=}k \}$

L		contexts ∈ $\Sigma^*\times\Sigma^*$					
		$(\varepsilon,\varepsilon)$	(a,ε)	(ε,c)	(a,c)	(a,bc)	(ab,c)
strings ∈ Σ^*	ε	✓	✓	✓	✓	✓	✓
	a	✓	✓			✓	
	b		✓	✓	✓		
	c	✓		✓			✓
	ab	✓		✓	✓	✓	
	bc	✓	✓		✓		✓
	abc	✓	✓	✓			
	$aabb$	✓		✓		✓	
	$bbcc$	✓	✓				✓

- $V = \{bc, abc\}$

- For $L \subseteq \Sigma^*$ and $V \subseteq \Sigma^*$, the **context set** of V is $L/V = \bigcap_{v \in V} L/v = \{\, w \in \Sigma^* \times \Sigma^* \mid w \odot V \subseteq L \,\}$
  - $U \subseteq V$ implies $L/U \supseteq L/V$

Example: $L = \{\, a^i b^j c^k \mid i{=}j \text{ or } j{=}k \,\}$

L		contexts $\in \Sigma^* \times \Sigma^*$					
		$(\varepsilon,\varepsilon)$	(a,ε)	(ε,c)	(a,c)	(a,bc)	(ab,c)
strings $\in \Sigma^*$	ε	✓	✓	✓	✓	✓	✓
	a	✓	✓			✓	
	b		✓	✓	✓		
	c	✓		✓			✓
	ab	✓		✓	✓	✓	
	bc	✓	✓		✓		✓
	abc	✓	✓	✓			
	$aabb$	✓		✓		✓	
	$bbcc$	✓	✓				✓

- $V = \{bc, abc\}$
- $L/V = \{(\varepsilon,\varepsilon), (a,\varepsilon), \ldots\}$

p-Finite Kernel Property

- Clark, Eyraud & Habrard ('08,'09), Yoshinaka ('11)
- $V \subseteq L(G,A)=L(A)$ is called a ***p*-kernel** of $L(A)$ (or of $A \in N$) iff $L(G)/V = L(G)/L(A)$ and $|V| \leqq p$

p-Finite Kernel Property

- Clark, Eyraud & Habrard ('08,'09), Yoshinaka ('11)
- $V \subseteq L(G,A)=L(A)$ is called a ***p*-kernel** of $L(A)$ (or of $A \in N$) iff $L(G)/V = L(G)/L(A)$ and $|V| \leqq p$
- CFG G has the ***p*-Finite Kernel Property (*p*-FKP)** iff every nonterminal admits a *p*-kernel

p-Finite Kernel Property

- Clark, Eyraud & Habrard ('08,'09), Yoshinaka ('11)
- $V \subseteq$ L(G,A)=L(A) is called a ***p*-kernel** of L(A) (or of $A \in N$)
  iff L(G)/V = L(G)/L(A) and | V | $\leqq p$
- CFG G has the ***p*-Finite Kernel Property (*p*-FKP)**
  iff every nonterminal admits a p-kernel
- L has the ***p*-FKP** if it is generated by a grammar with the p-FKP

p-Finite Kernel Property

- Clark, Eyraud & Habrard ('08,'09), Yoshinaka ('11)
- $V \subseteq$ L(G,A)=L(A) is called a ***p*-kernel** of L(A) (or of $A \in N$) iff L(G)/V = L(G)/L(A) and | V | ≦ p
- CFG G has the ***p*-Finite Kernel Property (*p*-FKP)** iff every nonterminal admits a p-kernel
- L has the ***p*-FKP** if it is generated by a grammar with the p-FKP
- 1-FKP is preserved under Chomsky-Normalization

p-Finite Kernel Property

- Clark, Eyraud & Habrard ('08,'09), Yoshinaka ('11)
- $V \subseteq$ L(G,A)=L(A) is called a ***p*-kernel** of L(A) (or of $A \in N$)
  iff L(G)/V = L(G)/L(A) and | V | ≦ p
- CFG G has the ***p*-Finite Kernel Property (*p*-FKP)**
  iff every nonterminal admits a p-kernel
- L has the ***p*-FKP** if it is generated by a grammar with the p-FKP
- 1-FKP is preserved under Chomsky-Normalization
- Every CFG with p-FKP has an equivalent one in CNF with q-FKP

- Every regular language has the 1-FKP
  - $\{ L/u \mid u \in \Sigma^* \}$ is finite iff L is regular
- Dyck language has the 1-FKP
- $\{ a^n b^n \mid n \geqq 0 \}$ has the 1-FKP
- $\{ a^n b^m \mid n \geqq m \}$ has the 1-FKP
- $\{ a^n b^n \mid n \geqq 0 \} \cup \{ a^n b^{2n} \mid n \geqq 0 \}$ has the 2-FKP
- Palindrome language has the 2-FKP
- $\{ a_1^{n1} a_2^{n2} \dots a_p^{np} \mid n_i = n_j \text{ for some } i \neq j \}$ separates the p-FKP from the $(p-1)$-FKP for $p \geqq 3$

- Palindrome = $\{ w \in \Sigma^* \mid w = w^R \}$
  admits a CFG with 2-FKP but no CFG with 1-FKP

- Palindrome = { $w \in \Sigma^*$ | $w = w^R$ }
  admits a CFG with 2-FKP but no CFG with 1-FKP

★ $S \rightarrow a | b | \varepsilon | S_A A \mid S_B B$, $S_A \rightarrow A\ S$, $S_B \rightarrow B\ S$, $A \rightarrow a$, $B \rightarrow b$

- Palindrome = $\{ w \in \Sigma^* \mid w = w^R \}$
  admits a CFG with 2-FKP but no CFG with 1-FKP

★ $S \to a \mid b \mid \varepsilon \mid S_A A \mid S_B B$, $S_A \to A\ S$, $S_B \to B\ S$, $A \to a$, $B \to b$

- $\{a\}$ is a 2-kernel of A by definition: Pal/$\{a\}$ = Pal/L(A)

- Palindrome = $\{ w \in \Sigma^* \mid w = w^R \}$
  admits a CFG with 2-FKP but no CFG with 1-FKP

★ $S \to a \mid b \mid \varepsilon \mid S_A A \mid S_B B$, $S_A \to A\ S$, $S_B \to B\ S$, $A \to a$, $B \to b$

- $\{a\}$ is a 2-kernel of A by definition: $\mathrm{Pal}/\{a\} = \mathrm{Pal}/\mathrm{L}(A)$

- $\{a,b\}$ is a 2-kernel of S, by $\mathrm{Pal}/\{a,b\} = \{(w,w^R) \mid w \in \Sigma^*\} = \mathrm{Pal}/\mathrm{L}(S)$.

- Palindrome = { $w \in \Sigma^*$ | $w = w^R$ }
  admits a CFG with 2-FKP but no CFG with 1-FKP

★ $S \rightarrow a|b|\varepsilon|\ S_A A \mid S_B B,\ \ S_A \rightarrow A\ S,\ \ S_B \rightarrow B\ S,\ A \rightarrow a,\ \ B \rightarrow b$

- {a} is a 2-kernel of A by definition: Pal/{a} = Pal/L(A)
- {a,b} is a 2-kernel of S, by Pal/{a,b} = {(w,w^R)| $w \in \Sigma^*$} = Pal/L(S) .
- {aa, ab} is a 2-kernel of S_A.

- Palindrome = $\{ w \in \Sigma^* \mid w = w^R \}$
  admits a CFG with 2-FKP but no CFG with 1-FKP

★ $S \to a | b | \varepsilon | S_A A \mid S_B B$, $S_A \to A\ S$, $S_B \to B\ S$, $A \to a$, $B \to b$

- $\{a\}$ is a 2-kernel of A by definition: Pal/$\{a\}$ = Pal/L(A)
- $\{a,b\}$ is a 2-kernel of S, by Pal/$\{a,b\}$ = $\{(w,w^R) \mid w \in \Sigma^*\}$ = Pal/L(S) .
- $\{aa, ab\}$ is a 2-kernel of S_A.

★ (ε, aw^R), $(\varepsilon, bw^R) \in$ Pal/$\{w\}$.
  Hence Pal/$\{a\}$ = Pal/V implies $V = \{w\}$.

- CFLs with the *p*-FKP are not identifiable in the limit from positive data only

- CFLs with the *p*-FKP are not identifiable in the limit from positive data only
- Any class that contains all finite languages and an infinite language is not identifiable in the limit from positive data

- CFLs with the *p*-FKP are not identifiable in the limit from positive data only
- Any class that contains all finite languages and an infinite language is not identifiable in the limit from positive data
- Why can one conjecture an infinite language from a finite subset of it, which may be the learning target itself?

- CFLs with the *p*-FKP are not identifiable in the limit from positive data only
- Any class that contains all finite languages and an infinite language is not identifiable in the limit from positive data
- Why can one conjecture an infinite language from a finite subset of it, which may be the learning target itself?
- Stronger learning scheme

Identification in the Limit from Positive Data and Membership Queries

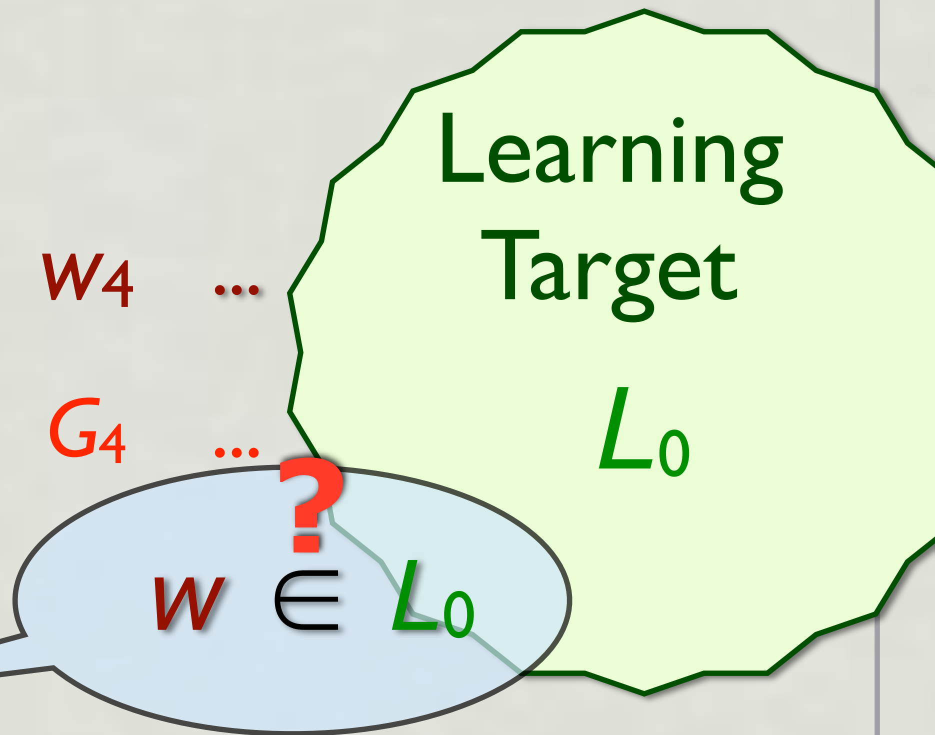
- Learner
  - gets a positive example w_1 w_2 w_3 w_4 ...
  - updates the conjecture G_1 G_2 G_3 G_4 ...
  - $L_0 = \{ w_1, w_2, w_3, \ldots \}$

Learning Target

L_0

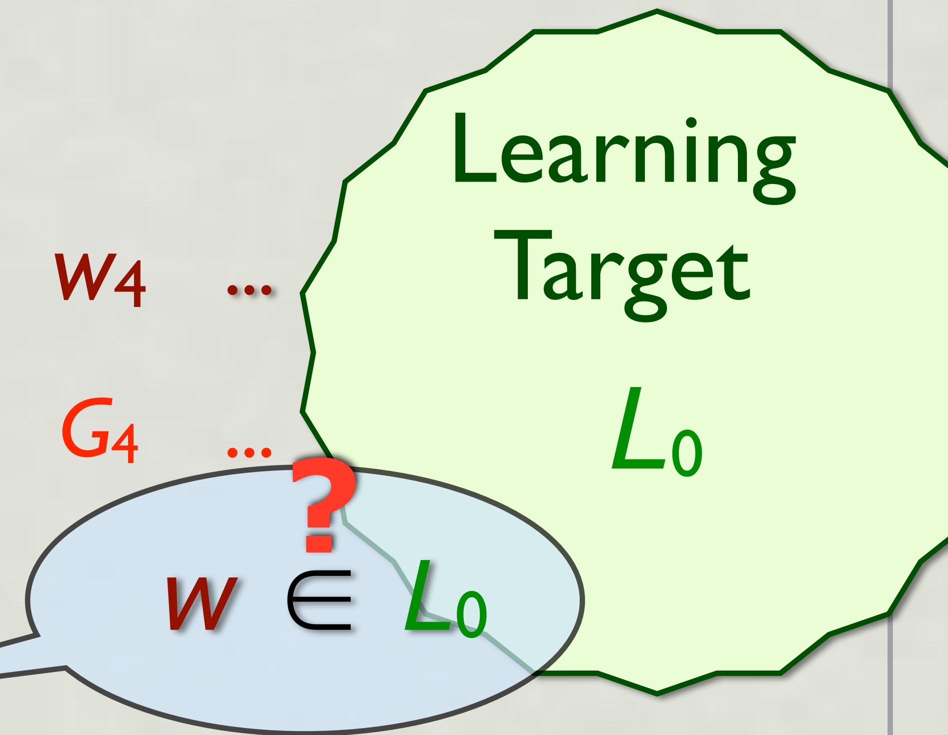
Identification in the Limit from Positive Data and Membership Queries

- Learner
  - gets a positive example w_1 w_2 w_3 w_4 ...
  - updates the conjecture G_1 G_2 G_3 G_4 ...
  - $L_0 = \{ w_1, w_2, w_3, \ldots \}$
  - calls the membership oracle



Identification in the Limit from Positive Data and Membership Queries

- Learner
  - gets a positive example w_1 w_2 w_3 w_4 ...
  - updates the conjecture G_1 G_2 G_3 G_4 ...
  - $L_0 = \{ w_1, w_2, w_3, \ldots \}$
  - calls the membership oracle

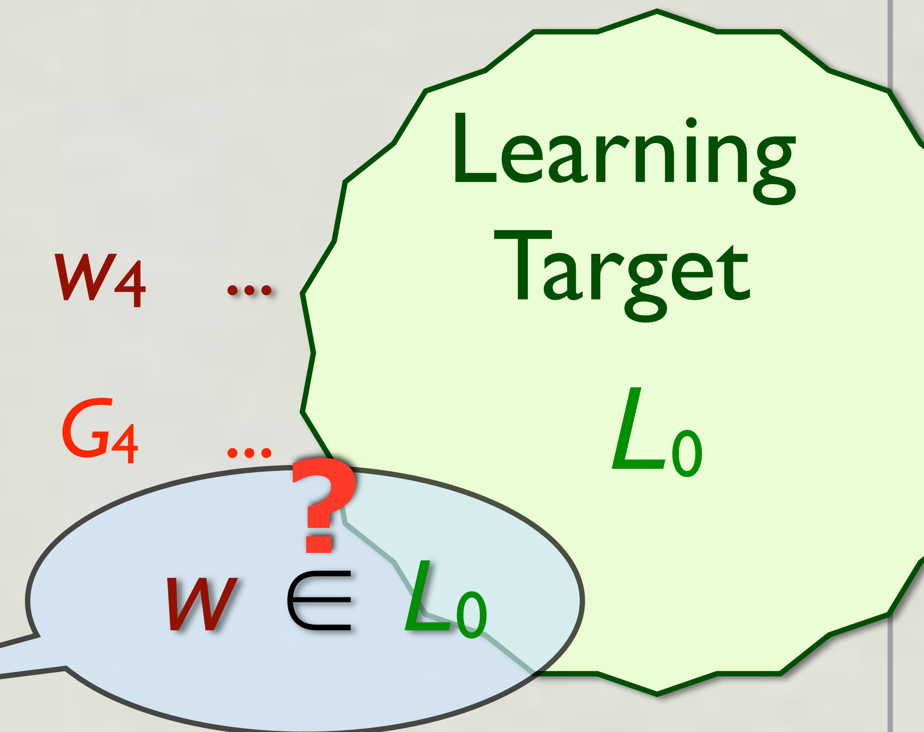


- Identification in the Limit:
  - convergence to a grammar for the target

    $G_n = G_{n+1} = G_{n+2} \ldots$ and $L(G_n) = L_0$

Identification in the Limit from Positive Data and Membership Queries

- Learner
  - gets a positive example w_1 w_2 w_3 w_4 ...
  - updates the conjecture G_1 G_2 G_3 G_4 ...
  - $L_0 = \{ w_1, w_2, w_3, \dots \}$
  - calls the membership oracle



- Identification in the Limit:
  - convergence to a grammar for the target

    $G_n = G_{n+1} = G_{n+2} \dots$ and $L(G_n) = L_0$

- Learner should uniformly learn a rich class of languages

Learner's Conjecture

- $K \subseteq \mathrm{Sub}(D)$ for $\{ v_1, ..., v_n \}$: positive data

Learner's Conjecture

- $K \subseteq \text{Sub}(D)$ for $\{ v_1, \ldots, v_n \}$: positive data
- $X = \text{Con}(D) = \{ w \mid w \odot v \in D \text{ for some } v \}$

Learner's Conjecture

- $K \subseteq \mathrm{Sub}(D)$ for $\{ v_1, \ldots, v_n \}$: positive data
- $X = \mathrm{Con}(D) = \{ w \mid w \odot v \in D$ for some $v \}$
- $G_{K,X}$: conjecture

Learner's Conjecture

- $K \subseteq \mathrm{Sub}(D)$ for $\{ v_1, \ldots, v_n \}$: positive data
- $X = \mathrm{Con}(D) = \{ w \mid w \odot v \in D \text{ for some } v \}$
- $G_{K,X}$: conjecture
  - $N = \{ [\![V]\!] \mid V \subseteq K, |V| \leqq p \}$

Learner's Conjecture

- $K \subseteq \mathrm{Sub}(D)$ for $\{ v_1, ..., v_n \}$: positive data
- $X = \mathrm{Con}(D) = \{ w \mid w \odot v \in D \text{ for some } v \}$
- $G_{K,X}$: conjecture
  - $N = \{ [\![V]\!] \mid V \subseteq K, |V| \leqq p \}$

★ $[\![V]\!] \Rightarrow u$ for $L_0/V \subseteq L_0/u$, i.e.,

i.e., $L_0/V \odot u \subseteq L_0$, i.e., $L_0/V = L_0/(V \cup \{u\})$

Learner's Conjecture

- $K \subseteq \mathrm{Sub}(D)$ for $\{ v_1, \ldots, v_n \}$: positive data
- $X = \mathrm{Con}(D) = \{ w \mid w \odot v \in D$ for some $v \}$
- $G_{K,X}$: conjecture
  - $N = \{ [\![V]\!] \mid V \subseteq K, |V| \leqq p \}$
  - ★ $[\![V]\!] \Rightarrow u$ for $L_0/V \subseteq L_0/u$, i.e.,
    i.e., $L_0/V \odot u \subseteq L_0$, i.e., $L_0/V = L_0/(V \cup \{u\})$
  - Initial Symbols: $\{ [\![V]\!] \in N \mid V \subseteq L_0 \}$

Learner's Conjecture

- $K \subseteq \mathrm{Sub}(D)$ for $\{ v_1, \ldots, v_n \}$: positive data
- $X = \mathrm{Con}(D) = \{ w \mid w \odot v \in D \text{ for some } v \}$
- $G_{K,X}$: conjecture
  - $N = \{ [\![V]\!] \mid V \subseteq K, |V| \leqq p \}$
  - ★ $[\![V]\!] \Rightarrow u$ for $L_0/V \subseteq L_0/u$, i.e.,

    i.e., $L_0/V \odot u \subseteq L_0$, i.e., $L_0/V = L_0/(V \cup \{u\})$
  - Initial Symbols: $\{ [\![V]\!] \in N \mid V \subseteq L_0 \}$
  - Rules

Learner's Conjecture

- $K \subseteq \mathrm{Sub}(D)$ for $\{ v_1, \ldots, v_n \}$: positive data
- $X = \mathrm{Con}(D) = \{ w \mid w \odot v \in D$ for some $v \}$
- $G_{K,X}$: conjecture
  - $N = \{ [\![V]\!] \mid V \subseteq K, |V| \leqq p \}$

  ★ $[\![V]\!] \Rightarrow u$ for $L_0/V \subseteq L_0/u$, i.e.,

  i.e., $L_0/V \odot u \subseteq L_0$, i.e., $L_0/V = L_0/(V \cup \{u\})$
  - Initial Symbols: $\{ [\![V]\!] \in N \mid V \subseteq L_0 \}$
  - Rules
    - Type I: $[\![V]\!] \rightarrow [\![V_1]\!][\![V_2]\!]$ if $L_0/V \cap X \subseteq L_0/(V_1 V_2) \cap X$

Learner's Conjecture

- $K \subseteq \mathrm{Sub}(D)$ for $\{ v_1, \ldots, v_n \}$: positive data
- $X = \mathrm{Con}(D) = \{ w \mid w \odot v \in D$ for some $v \}$
- $G_{K,X}$: conjecture
    - $N = \{ [\![V]\!] \mid V \subseteq K, |V| \leqq p \}$
    - ★ $[\![V]\!] \Rightarrow u$ for $L_0/V \subseteq L_0/u$, i.e.,
      i.e., $L_0/V \odot u \subseteq L_0$, i.e., $L_0/V = L_0/(V \cup \{u\})$
    - Initial Symbols: $\{ [\![V]\!] \in N \mid V \subseteq L_0 \}$
    - Rules
        - Type I: $[\![V]\!] \to [\![V_1]\!][\![V_2]\!]$ if $L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$
        - Type II: $[\![V]\!] \to a$ if $L_0/V \cap X \subseteq L_0/a \cap X$

- Type I: $[\![V]\!] \rightarrow [\![V_1]\!][\![V_2]\!]$ if $L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$

  - ★ $[\![V]\!] \Rightarrow u$ for $L_0/V \subseteq L_0/u$

- Type I: $[\![V]\!] \to [\![V_1]\!][\![V_2]\!]$ if $L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$

  ★ $[\![V]\!] \Rightarrow u$ for $L_0/V \subseteq L_0/u$

- $L_0/V \subseteq L_0/(V_1V_2)$ ⟶ $L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$

- Type I: $\llbracket V \rrbracket \to \llbracket V_1 \rrbracket \llbracket V_2 \rrbracket$ if $L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$

  ★ $\llbracket V \rrbracket \Rightarrow u$ for $L_0/V \subseteq L_0/u$

- $L_0/V \subseteq L_0/(V_1V_2) \quad \underset{\not\Leftarrow}{\Rightarrow} \quad L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$

- Type I: $[\![V]\!] \to [\![V_1]\!][\![V_2]\!]$ if $L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$

  ★ $[\![V]\!] \Rightarrow u$ for $L_0/V \subseteq L_0/u$

- $L_0/V \subseteq L_0/(V_1V_2) \quad \Rightarrow,\ \not\Leftarrow \quad L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$

- $[\![V]\!] \to [\![V_1]\!][\![V_2]\!]$ is **incorrect** iff $L_0/V \not\subseteq L_0/(V_1V_2)$
  $[\![V]\!] \to a$ is **incorrect** iff $L_0/V \not\subseteq L_0/a$

- Type I: $[\![V]\!] \to [\![V_1]\!][\![V_2]\!]$ if $L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$

  ★ $[\![V]\!] \Rightarrow u$ for $L_0/V \subseteq L_0/u$

- $L_0/V \subseteq L_0/(V_1V_2) \;\; \overset{\Longrightarrow}{\not\Longleftarrow} \;\; L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$

- $[\![V]\!] \to [\![V_1]\!][\![V_2]\!]$ is **incorrect** iff $L_0/V \not\subseteq L_0/(V_1V_2)$
  $[\![V]\!] \to a$ is **incorrect** iff $L_0/V \not\subseteq L_0/a$

- If X is rich enough w.r.t. K, $G_{K,X}$ has no incorrect rules

- Type I: $[\![V]\!] \to [\![V_1]\!][\![V_2]\!]$ if $L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$

  ★ $[\![V]\!] \Rightarrow u$ for $L_0/V \subseteq L_0/u$

- $L_0/V \subseteq L_0/(V_1V_2)$ $\Rightarrow$ / $\not\Leftarrow$ $L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$

- $[\![V]\!] \to [\![V_1]\!][\![V_2]\!]$ is **incorrect** iff $L_0/V \not\subseteq L_0/(V_1V_2)$
  $[\![V]\!] \to a$ is **incorrect** iff $L_0/V \not\subseteq L_0/a$

- If X is rich enough w.r.t. K, $G_{K,X}$ has no incorrect rules

- X is **fiducial** on K, iff
  for any $V, V_1, V_2 \in K$ with $L_0/V \not\subseteq L_0/(V_1V_2)$
  there is $w \in (L_0/V - L_0/(V_1V_2)) \cap X$
  and
  $\forall V \in K, a \in \Sigma$ with $L_0/V \not\subseteq L_0/a$, $\exists w \in (L_0/V - L_0/a) \cap X$

- Type I: $[\![V]\!] \to [\![V_1]\!][\![V_2]\!]$ if $L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$

  ★ $[\![V]\!] \Rightarrow u$ for $L_0/V \subseteq L_0/u$

- $L_0/V \subseteq L_0/(V_1V_2)$ $\Rightarrow$ / $\not\Leftarrow$ $L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$

- $[\![V]\!] \to [\![V_1]\!][\![V_2]\!]$ is **incorrect** iff $L_0/V \not\subseteq L_0/(V_1V_2)$
  $[\![V]\!] \to a$ is **incorrect** iff $L_0/V \not\subseteq L_0/a$

- If X is rich enough w.r.t. K, $G_{K,X}$ has no incorrect rules

- X is **fiducial** on K, iff
  for any $V, V_1, V_2 \in K$ with $L_0/V \not\subseteq L_0/(V_1V_2)$
  there is $w \in (L_0/V - L_0/(V_1V_2)) \cap X$
  and
  $\forall\, V \in K, a \in \Sigma$ with $L_0/V \not\subseteq L_0/a$, $\exists\, w \in (L_0/V - L_0/a) \cap X$

- X is fiducial on K iff the conjecture has no incorrect rule

Monotonicity

- $G_{K,X}$: conjecture
  - $N = \{ [\![V]\!] \mid V \subseteq K, |V| \leqq p \}$
  - Initial Symbols: $\{ [\![V]\!] \in N \mid V \subseteq L_0 \}$
  - Rules
    - Type I: $[\![V]\!] \rightarrow [\![V_1]\!][\![V_2]\!]$ if $L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$
    - Type II: $[\![V]\!] \rightarrow a$ if $L_0/V \cap X \subseteq L_0/a \cap X$

Expanding K $\Rightarrow$ More nonterminals & rules

Expanding X $\Rightarrow$ Less incorrect rules

Algorithm

Let D := K := X := ∅; G := vacuous grammar;
For i = 1,2,3,...
  let D := $\{w_1, w_2, \ldots, w_i\}$;
  If $D \not\subseteq L(G)$
    then let K := Sub(D);
  End if
  let X := Con(D);
  update G with K and X;
End for

Completeness

- $G_{K,X}$: conjecture, G_0 : grammar in CNF s.t. $L_0 = L(G_0)$

Completeness

- $G_{K,X}$: conjecture, G_0 : grammar in CNF s.t. $L_0 = \mathrm{L}(G_0)$
- A: nonterminal of G_0

Completeness

- $G_{K,X}$: conjecture, G_0 : grammar in CNF s.t. $L_0 = \mathrm{L}(G_0)$
- A: nonterminal of G_0
- V_A : a kernel of A: $L_0/V_A = L_0/\mathrm{L}(G_0,A)$

Completeness

- $G_{K,X}$: conjecture, G_0 : grammar in CNF s.t. $L_0 = L(G_0)$
- A: nonterminal of G_0
- V_A : a kernel of A: $L_0/V_A = L_0/L(G_0,A)$
- If $V_A \subseteq K$ for all nonterminal A of G_0, then $G_0 \subseteq L(G_{K,X})$.

Completeness

- $G_{K,X}$: conjecture, G_0 : grammar in CNF s.t. L_0 = L(G_0)
- A: nonterminal of G_0
- V_A : a kernel of A: $L_0/V_A = L_0/\mathrm{L}(G_0,A)$
- If $V_A \subseteq K$ for all nonterminal A of G_0, then $G_0 \subseteq \mathrm{L}(G_{K,X})$.
  - For $A \rightarrow BC$ in G_0 ,
    $[\![V_A]\!] \rightarrow [\![V_B]\!][\![V_C]\!]$ (Type I) in $G_{K,X}$
    by $L_0/\mathrm{L}(A) = L_0/V_A \subseteq L_0/(V_B V_C) = L_0/\mathrm{L}(BC)$.

Completeness

- $G_{K,X}$: conjecture, G_0 : grammar in CNF s.t. $L_0 = \mathrm{L}(G_0)$
- A: nonterminal of G_0
- V_A : a kernel of A: $L_0/V_A = L_0/\mathrm{L}(G_0,A)$
- If $V_A \subseteq K$ for all nonterminal A of G_0, then $G_0 \subseteq \mathrm{L}(G_{K,X})$.
  - For $A \to BC$ in G_0 ,
    $[\![V_A]\!] \to [\![V_B]\!][\![V_C]\!]$ (Type I) in $G_{K,X}$
    by $L_0/\mathrm{L}(A) = L_0/V_A \subseteq L_0/(V_B V_C) = L_0/\mathrm{L}(BC)$.
  - For $A \to a$ in G_0 ,
    $[\![V_A]\!] \to a$ (Type II) in $G_{K,X}$.

Completeness

- $G_{K,X}$: conjecture, G_0 : grammar in CNF s.t. $L_0 = \mathrm{L}(G_0)$
- A: nonterminal of G_0
- V_A : a kernel of A: $L_0/V_A = L_0/\mathrm{L}(G_0,A)$
- If $V_A \subseteq K$ for all nonterminal A of G_0, then $G_0 \subseteq \mathrm{L}(G_{K,X})$.
  - For $A \to BC$ in G_0,
    $[\![V_A]\!] \to [\![V_B]\!][\![V_C]\!]$ (Type I) in $G_{K,X}$
    by $L_0/\mathrm{L}(A) = L_0/V_A \subseteq L_0/(V_BV_C) = L_0/\mathrm{L}(BC)$.
  - For $A \to a$ in G_0,
    $[\![V_A]\!] \to a$ (Type II) in $G_{K,X}$.
- $[\![V_S]\!]$ is an initial symbol for $V_S \in L$

Soundness

- $G_{K,X}$: conjecture

Soundness

- $G_{K,X}$: conjecture
- Proposition: $L(G_{K,X}) \subseteq L_0$ if X is fiducial on K.

Soundness

- $G_{K,X}$: conjecture
- Proposition: $L(G_{K,X}) \subseteq L_0$ if X is fiducial on K.
  - If $[\![V]\!] \Rightarrow u$ in $G_{K,X}$, then $L_0/V \subseteq L_0/u$

Soundness

- $G_{K,X}$: conjecture
- Proposition: $L(G_{K,X}) \subseteq L_0$ if X is fiducial on K.
  - If $[\![V]\!] \Rightarrow u$ in $G_{K,X}$, then $L_0/V \subseteq L_0/u$
  - Suppose $[\![V]\!] \Rightarrow [\![V_1]\!]\,[\![V_2]\!] \Rightarrow u_1 u_2$ in $G_{K,X}$. By i.h., $L_0/V_i \subseteq L_0/u_i$.

Soundness

- $G_{K,X}$: conjecture
- Proposition: $L(G_{K,X}) \subseteq L_0$ if X is fiducial on K.
  - If $[\![V]\!] \Rightarrow u$ in $G_{K,X}$, then $L_0/V \subseteq L_0/u$
  - Suppose $[\![V]\!] \Rightarrow [\![V_1]\!][\![V_2]\!] \Rightarrow u_1u_2$ in $G_{K,X}$. By i.h., $L_0/V_i \subseteq L_0/u_i$.
    - Type I: $[\![V]\!] \rightarrow [\![V_1]\!][\![V_2]\!]$
      if $L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$, i.e., $L_0/V \subseteq L_0/(V_1V_2)$

Soundness

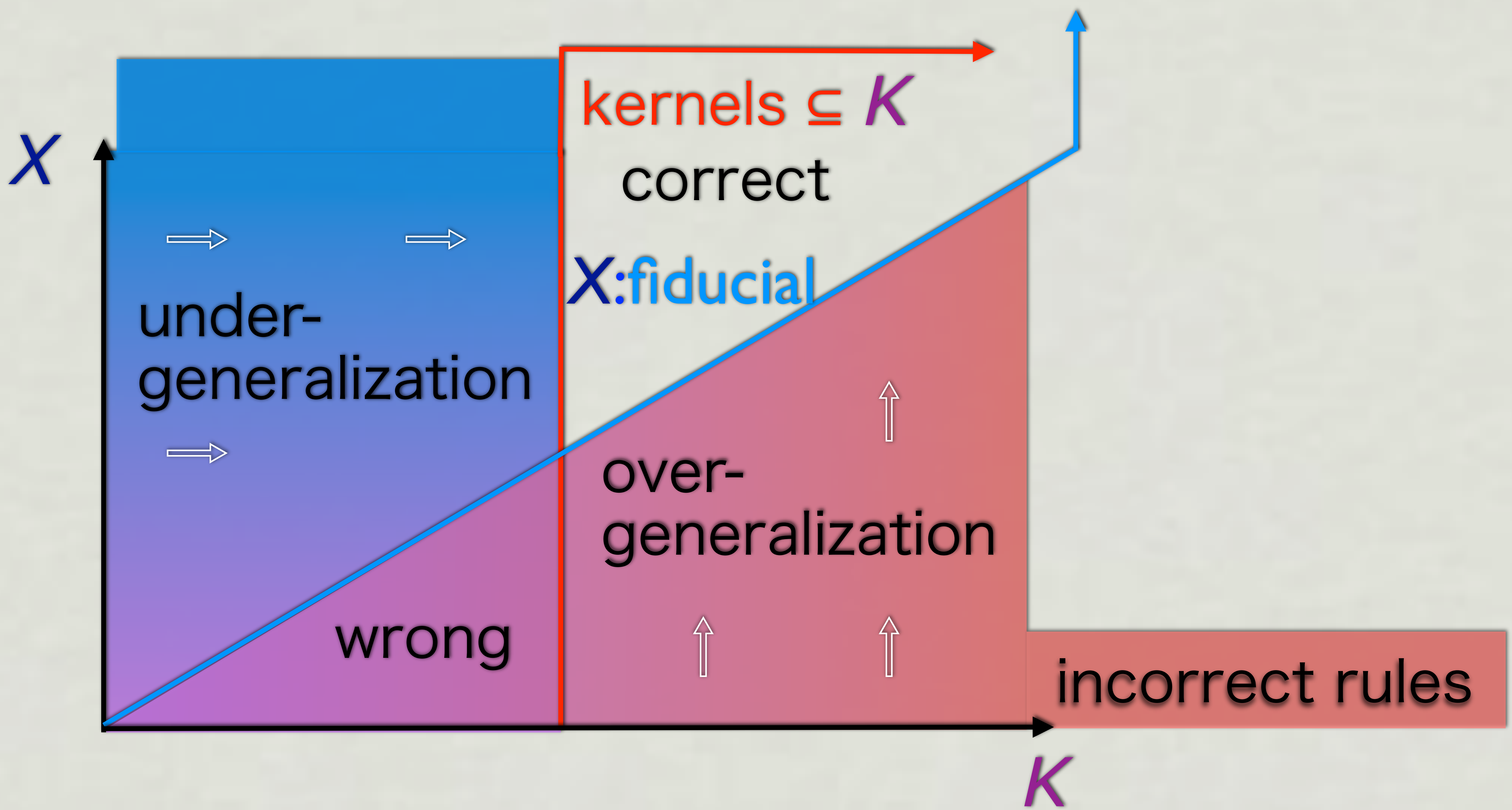
- $G_{K,X}$: conjecture
- Proposition: $L(G_{K,X}) \subseteq L_0$ if X is fiducial on K.
  - If $[\![V]\!] \Rightarrow u$ in $G_{K,X}$, then $L_0/V \subseteq L_0/u$
  - Suppose $[\![V]\!] \Rightarrow [\![V_1]\!][\![V_2]\!] \Rightarrow u_1u_2$ in $G_{K,X}$. By i.h., $L_0/V_i \subseteq L_0/u_i$.
    - Type I: $[\![V]\!] \to [\![V_1]\!][\![V_2]\!]$
      if $L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$, i.e., $L_0/V \subseteq L_0/(V_1V_2)$
  - For $w \in L_0/V \subseteq L_0/(V_1V_2)$,
    $w \odot V_1V_2 \subseteq L_0 \longrightarrow w \odot u_1V_2 \subseteq L_0 \longrightarrow w \odot u_1u_2 \subseteq L_0$,
    i.e., $w \in L_0/u_1u_2$.

Soundness

- $G_{K,X}$: conjecture
- Proposition: $L(G_{K,X}) \subseteq L_0$ if X is fiducial on K.
  - If $[\![V]\!] \Rightarrow u$ in $G_{K,X}$, then $L_0/V \subseteq L_0/u$
  - Suppose $[\![V]\!] \Rightarrow [\![V_1]\!][\![V_2]\!] \Rightarrow u_1u_2$ in $G_{K,X}$. By i.h., $L_0/V_i \subseteq L_0/u_i$.
    - Type I: $[\![V]\!] \to [\![V_1]\!][\![V_2]\!]$
      if $L_0/V \cap X \subseteq L_0/(V_1V_2) \cap X$, i.e., $L_0/V \subseteq L_0/(V_1V_2)$
  - For $w \in L_0/V \subseteq L_0/(V_1V_2)$,
    $w \odot V_1V_2 \subseteq L_0 \longrightarrow w \odot u_1V_2 \subseteq L_0 \longrightarrow w \odot u_1u_2 \subseteq L_0$,
    i.e., $w \in L_0/u_1u_2$.
  - For a start symbol $[\![V]\!]$, $(\varepsilon,\varepsilon) \in L_0/V$.

Convergence

- Conjecture is not updated infinitely many times
  - K will contain a p-kernel for each nonterminal of G_0
  - $\Rightarrow$ K will be converged
  - X will be fiducial on K
  - Infinite update of X does not update our conjecture


kernels ⊆ K
X
correct
X:fiducial
under-
generalization
wrong
over-
generalization
incorrect rules
K

Theorem

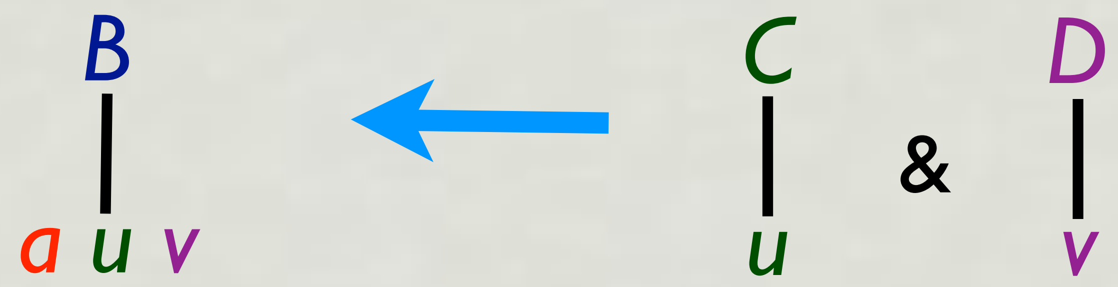
- The algorithm identifies every CFL in CNF with the *p*-FKP in the limit from positive data & membership queries
- Polynomial-time update (Polynomial number of MQs)
- Polynomially many examples are enough for convergence w.r.t. the size of the grammar to be learnt

Distributional Learning of Multiple Context-Free Grammars

Multiple CFGs

- $B \to a\ C\ D$ (context-free rule)

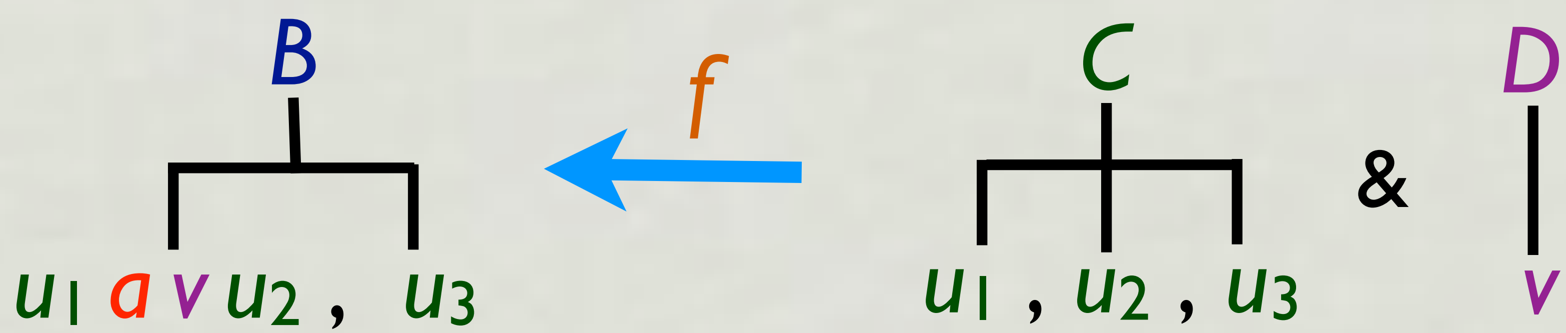
★ a : terminal symbol



- $B \to g(C,D)$

★ $g(x, y) = axy$

- $B \to f(C,D)$ (multiple cf rule)

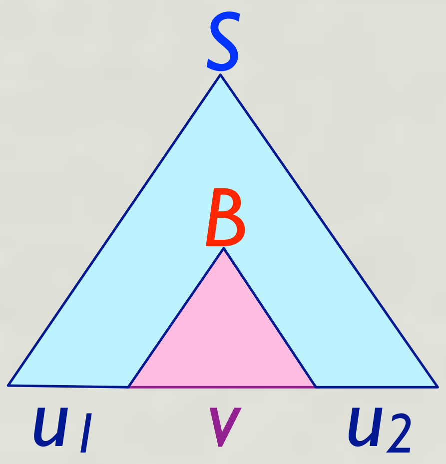


★ $\dim(B) = 2$, $\dim(C) = 3$, $\dim(D) = 1$.

★ $f(\langle x_1,x_2,x_3\rangle, \langle y\rangle) = \langle x_1ayx_2, x_3\rangle$

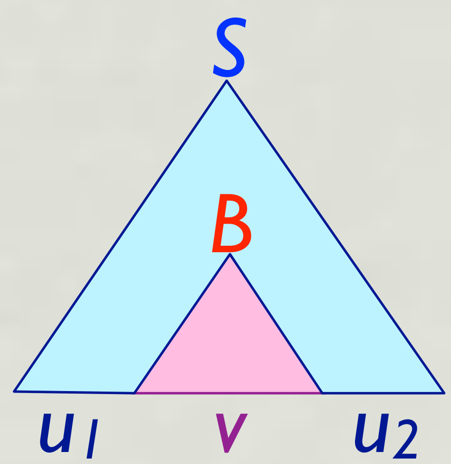
$$B\langle x_1ayx_2, x_3\rangle :\!- \ C\langle x_1,x_2,x_3\rangle, D\langle y\rangle$$

- CFG learning $w = u_1 v u_2 \in D$



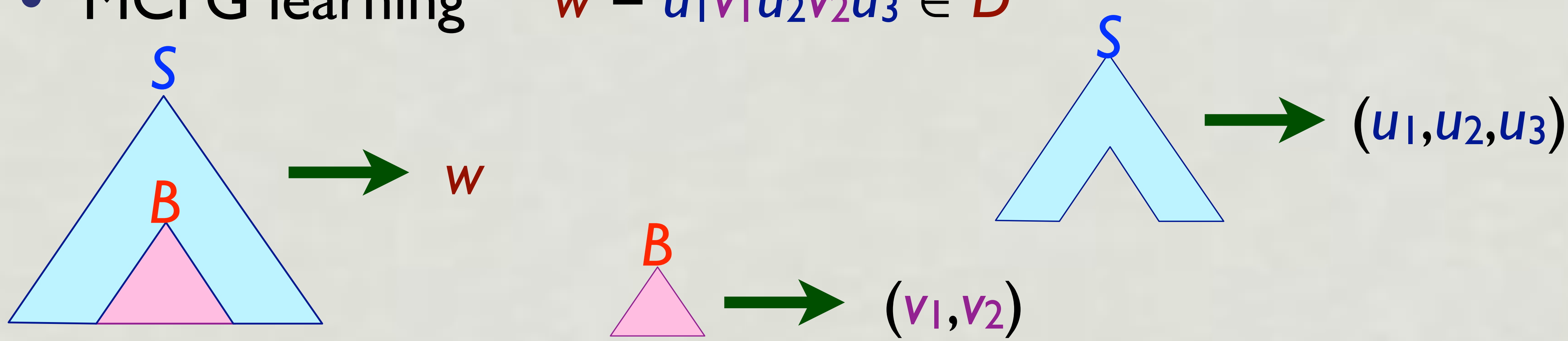
- Sub(D) = $\{ v \mid u_1 v u_2 \in D$ for some $u_1, u_2 \}$
- Con(D) = $\{ (u_1, u_2) \mid u_1 v u_2 \in D$ for some $v \}$

- CFG learning $w = u_1 v u_2 \in D$



- Sub(D) = { v | $u_1 v u_2 \in D$ for some u_1, u_2 }
- Con(D) = { (u_1, u_2) | $u_1 v u_2 \in D$ for some v }

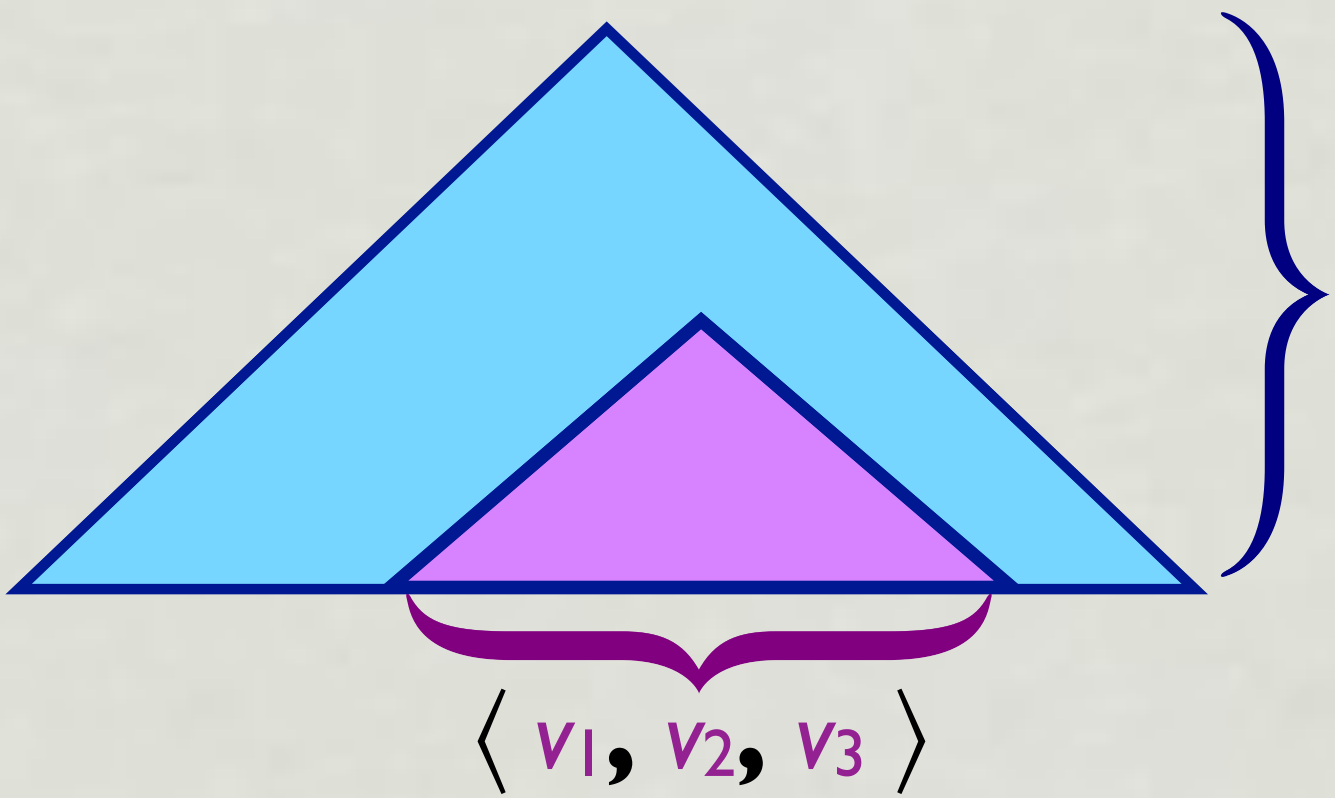
- MCFG learning $w = u_1 v_1 u_2 v_2 u_3 \in D$



- Sub$_1$(D) = { v | $u_1 v u_2 \in D$ for some u_1, u_2 }
  Con$_1$(D) = { (u_1, u_2) | $u_1 v u_2 \in D$ for some v }
- Sub$_2$(D) = { (v_1, v_2) | $u_1 v_1 u_2 v_2 u_3 \in D$ for some u_1, u_2, u_3 }
  Con$_2$(D) = { (u_1, u_2, u_3) | $u_1 v_1 u_2 v_2 u_3 \in D$ for some v_1, v_2 },
- and so on

Non-Erasing & Non-Permuting

- $f(\mathbf{z}_1,\ldots,\mathbf{z}_n) = \langle t_1,\ldots,t_{\dim(A)}\rangle$
  - Non-Erasing:
    Each variable $z_{i,j}$ occurs just once in $t_1,\ldots,t_{\dim(A)}$
  - Non-Permuting:
    Variables $z_{i,1},\ldots,z_{i,\dim(B_i)}$ occur in this order in $t_1,\ldots,t_{\dim(A)}$



$\langle \ldots, \ldots v_1 \ldots v_2 \ldots, \ldots v_3 \ldots, \ldots \rangle$

✔ $f(\langle x_1,x_2,x_3\rangle, \langle y\rangle) = \langle x_1 a y x_2, x_3\rangle$

✖ $f(\langle x_1,x_2,x_3\rangle, \langle y\rangle) = \langle x_1 a y x_3, x_2\rangle$

Notations

Notations

- *m*-MCFG(*n*):
  - dim(*A*) ≦ *m* for all *A* (*A* generates dim(*A*)-tuples)
  - each rule has at most *n* nonterminals on the rhs

Notations

- m-MCFG(n):
  - dim(A) ≦ m for all A (A generates dim(A)-tuples)
  - each rule has at most n nonterminals on the rhs
- $w_0 v_1 w_1 \ldots v_m w_m = \langle w_0\_w_1\_\ldots\_w_m \rangle \odot \langle v_1, \ldots, v_m \rangle = \boldsymbol{w} \odot \boldsymbol{v}$

Notations

- m-MCFG(n):
  - dim(A) ≦ m for all A (A generates dim(A)-tuples)
  - each rule has at most n nonterminals on the rhs
- $w_0 v_1 w_1 \ldots v_m w_m = \langle w_0\_w_1\_\ldots\_w_m \rangle \odot \langle v_1, \ldots, v_m \rangle = \boldsymbol{w} \odot \boldsymbol{v}$
- *Multi-context*: tuple of strings $\boldsymbol{w} = \langle w_0\_w_1\_\ldots\_w_m \rangle$

Notations

- m-MCFG(n):
  - dim(A) ≦ m for all A (A generates dim(A)-tuples)
  - each rule has at most n nonterminals on the rhs
- $w_0 v_1 w_1 \ldots v_m w_m = \langle w_0\_w_1\_\ldots\_w_m \rangle \odot \langle v_1, \ldots, v_m \rangle = \boldsymbol{w} \odot \boldsymbol{v}$
- *Multi-context*: tuple of strings $\boldsymbol{w} = \langle w_0\_w_1\_\ldots\_w_m \rangle$
- *Multi-word*: tuple of strings $\boldsymbol{v} = \langle v_1, v_2, \ldots, v_m \rangle$

Notations

- m-MCFG(n):
  - dim(A) ≦ m for all A (A generates dim(A)-tuples)
  - each rule has at most n nonterminals on the rhs
- $w_0 v_1 w_1 \ldots v_m w_m = \langle w_0\_w_1\_\ldots\_w_m \rangle \odot \langle v_1, \ldots, v_m \rangle = \boldsymbol{w} \odot \boldsymbol{v}$
- *Multi-context*: tuple of strings $\boldsymbol{w} = \langle w_0\_w_1\_\ldots\_w_m \rangle$
- *Multi-word*: tuple of strings $\boldsymbol{v} = \langle v_1, v_2, \ldots, v_m \rangle$
- $L_0/\boldsymbol{v} = \{ \boldsymbol{w} \mid \boldsymbol{w} \odot \boldsymbol{v} \in L_0 \}$

  e.g. $a^*b^*c^*/\langle ab, cc \rangle = \langle a^*\_b^*c^*\_c^* \rangle$

Easy Lemma

- If $L/\boldsymbol{v}_i \subseteq L/\boldsymbol{u}_i$ for $i = 1,\ldots,k$, then
  $L/f(\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k) \subseteq L/f(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k)$ for any non-erasing non-permuting f

Easy Lemma

- If $L/\mathbf{v}_i \subseteq L/\mathbf{u}_i$ for $i = 1,\ldots,k$, then $L/f(\mathbf{v}_1,\ldots,\mathbf{v}_k) \subseteq L/f(\mathbf{u}_1,\ldots,\mathbf{u}_k)$ for any non-erasing non-permuting f

  $\because$ Let $\mathbf{w} \in f(\mathbf{v}_1,\ldots,\mathbf{v}_k)$.

  $\mathbf{w} \odot f(\mathbf{v}_1,\ldots,\mathbf{v}_k) \in L \;\Rightarrow\; \mathbf{w} \odot f(\mathbf{u}_1,\mathbf{v}_2,\ldots,\mathbf{v}_k) \in L$

  $\Rightarrow\; \mathbf{w} \odot f(\mathbf{u}_1,\mathbf{u}_2,\mathbf{v}_3,\ldots,\mathbf{v}_k) \in L \;\Rightarrow\; \ldots \;\Rightarrow\; \mathbf{w} \odot f(\mathbf{u}_1,\ldots,\mathbf{u}_k) \in L$.

Easy Lemma

- If $L/\mathbf{v}_i \subseteq L/\mathbf{u}_i$ for $i = 1,\ldots,k$, then $L/f(\mathbf{v}_1,\ldots,\mathbf{v}_k) \subseteq L/f(\mathbf{u}_1,\ldots,\mathbf{u}_k)$ for any non-erasing non-permuting f

  $\because$ Let $\mathbf{w} \in f(\mathbf{v}_1,\ldots,\mathbf{v}_k)$.

  $\mathbf{w} \odot f(\mathbf{v}_1,\ldots,\mathbf{v}_k) \in L \;\Rightarrow\; \mathbf{w} \odot f(\mathbf{u}_1,\mathbf{v}_2,\ldots,\mathbf{v}_k) \in L$

  $\Rightarrow\; \mathbf{w} \odot f(\mathbf{u}_1,\mathbf{u}_2,\mathbf{v}_3,\ldots,\mathbf{v}_k) \in L \;\Rightarrow\; \ldots \;\Rightarrow\; \mathbf{w} \odot f(\mathbf{u}_1,\ldots,\mathbf{u}_k) \in L$.

- If $L/\mathbf{v}_i = L/\mathbf{u}_i$ for $i = 1,\ldots,k$, then $L/f(\mathbf{v}_1,\ldots,\mathbf{v}_k) = L/f(\mathbf{u}_1,\ldots,\mathbf{u}_k)$ for any non-erasing non-permuting f

m-dimensionally substitutable

- L is mD-**substitutable** iff $L/\mathbf{v}_1 \cap L/\mathbf{v}_2 \neq \varnothing$ implies $L/\mathbf{v}_1 = L/\mathbf{v}_2$ for any $\mathbf{v}_1$, $\mathbf{v}_2$ with $|\mathbf{v}_1| = |\mathbf{v}_2| \leqq m$

m-dimensionally substitutable

- L is *m*D-**substitutable** iff $L/\mathbf{v}_1 \cap L/\mathbf{v}_2 \neq \varnothing$ implies $L/\mathbf{v}_1 = L/\mathbf{v}_2$ for any $\mathbf{v}_1, \mathbf{v}_2$ with $|\mathbf{v}_1| = |\mathbf{v}_2| \leqq m$
- Substitutability of a CFL = 1D-substitutability

m-dimensionally substitutable

- L is mD-**substitutable** iff $L/\mathbf{v}_1 \cap L/\mathbf{v}_2 \neq \varnothing$ implies $L/\mathbf{v}_1 = L/\mathbf{v}_2$ for any $\mathbf{v}_1, \mathbf{v}_2$ with $|\mathbf{v}_1| = |\mathbf{v}_2| \leqq m$
- Substitutability of a CFL = 1D-substitutability
- ex. $L = \{ a^n bc^n de^n \mid n \geqq 0 \}$ is 2D-substitutable.
  (abc,e) and ($aabcc$, ee) share multi-context (ε_d_ε).

(ε_d_ε) ⊙ (abc,e) = $abcde$

(ε_d_ε) ⊙ ($aabcc$,ee) = $aabccdee$

m-dimensionally substitutable

- L is mD-**substitutable** iff $L/\mathbf{v}_1 \cap L/\mathbf{v}_2 \neq \varnothing$ implies $L/\mathbf{v}_1 = L/\mathbf{v}_2$ for any $\mathbf{v}_1, \mathbf{v}_2$ with $|\mathbf{v}_1| = |\mathbf{v}_2| \leqq m$
- Substitutability of a CFL = 1D-substitutability
- ex. $L = \{ a^n bc^n de^n \mid n \geqq 0 \}$ is 2D-substitutable.
  (abc,e) and ($aabcc$, ee) share multi-context ($\varepsilon\_d\_\varepsilon$).

  ($\varepsilon\_d\_\varepsilon$) $\odot$ (abc,e) = $abcde$

  ($\varepsilon\_d\_\varepsilon$) $\odot$ ($aabcc$,ee) = $aabccdee$
- Learning Target: mD-substitutable m-MCFL(n)

m-dimensionally substitutable

- L is mD-**substitutable** iff $L/\mathbf{v}_1 \cap L/\mathbf{v}_2 \neq \varnothing$ implies $L/\mathbf{v}_1 = L/\mathbf{v}_2$ for any $\mathbf{v}_1$, $\mathbf{v}_2$ with $|\mathbf{v}_1| = |\mathbf{v}_2| \leqq m$
- Substitutability of a CFL = 1D-substitutability
- ex. $L = \{ a^n bc^n de^n \mid n \geqq 0 \}$ is 2D-substitutable.
  (abc,e) and ($aabcc$, ee) share multi-context (ε_d_ε).

  (ε_d_ε) ⊙ (abc,e) = $abcde$

  (ε_d_ε) ⊙ ($aabcc$,ee) = $aabccdee$
- Learning Target: mD-substitutable m-MCFL(n)
- Identification in the limit from positive data

Learning *m*D-substitutable MCFLs

```
let G := vacuous grammar;
For n = 1,2,3,...
  let D := {w1 , w2 , ... , wn };
  If D ⊈ L(G)
    then update G by D;
  End if
  output G
End for
```

Learner's Conjecture

- CFG Learning

- $N = \mathrm{Sub}_1(D)$

★ $[\![v]\!] \Rightarrow u$ for $L/v = L/u$,

- Initial Symbols: $\{\ [\![v]\!] \in N \mid v \in D\ \}$
- Rules
  - $[\![v_1 v_2]\!] \rightarrow [\![v_1]\!][\![v_2]\!]$, $[\![a]\!] \rightarrow a$
  - $[\![v]\!] \rightarrow [\![u]\!]$ if $\exists\, w$ s.t. $w \odot v, w \odot u \in D$

Learner's Conjecture

- CFG Learning
  - $N = \mathrm{Sub}_1(D)$
  - ★ $[\![v]\!] \Rightarrow u$ for $L/v = L/u$,
  - Initial Symbols: $\{ [\![v]\!] \in N \mid v \in D \}$
  - Rules
    - $[\![v_1 v_2]\!] \to [\![v_1]\!][\![v_2]\!]$, $[\![a]\!] \to a$
    - $[\![v]\!] \to [\![u]\!]$ if $\exists\, w$ s.t. $w \odot v, w \odot u \in D$

- MCFG Learning
  - $N = N_1 \cup \ldots \cup N_m$, where $N_k = \mathrm{Sub}_k(D)$ for $k \leqq m$
  - ★ $[\![\mathbf{v}]\!] \Rightarrow \mathbf{u}$ for $L/\mathbf{v} = L/\mathbf{u}$,
  - Initial Symbols: $\{ [\![v]\!] \in N_1 \mid v \in D \}$
  - $[\![\mathbf{v}_0]\!] \to f([\![\mathbf{v}_1]\!],\ldots,[\![\mathbf{v}_k]\!])$ for $\mathbf{v}_0 = f(\mathbf{v}_1,\ldots,\mathbf{v}_k)$ with $k \leqq n$
  - $[\![\mathbf{v}]\!] \to [\![\mathbf{u}]\!]$ if $\exists\, \mathbf{w}$ s.t. $\mathbf{w} \odot \mathbf{v}, \mathbf{w} \odot \mathbf{u} \in D$

Soundness

- G_D : conjecture

Soundness

- G_D : conjecture
- Proposition: $L(G_D) \subseteq L_0$

Soundness

- G_D : conjecture
- Proposition: $L(G_D) \subseteq L_0$
  - If $[\![v]\!] \Rightarrow u$ in G_D, then $L_0/v = L_0/u$

Soundness

- G_D : conjecture
- Proposition: $L(G_D) \subseteq L_0$
  - If $[\![\boldsymbol{v}]\!] \Rightarrow \boldsymbol{u}$ in G_D, then $L_0/\boldsymbol{v} = L_0/\boldsymbol{u}$
  - $[\![\boldsymbol{v}_0]\!] \to f([\![\boldsymbol{v}_1]\!],\ldots,[\![\boldsymbol{v}_k]\!])$ for $f(\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k) = \boldsymbol{v}_0$

Soundness

- G_D : conjecture
- Proposition: $L(G_D) \subseteq L_0$
  - If $[\![\boldsymbol{v}]\!] \Rightarrow \boldsymbol{u}$ in G_D, then $L_0/\boldsymbol{v} = L_0/\boldsymbol{u}$
  - $[\![\boldsymbol{v}_0]\!] \rightarrow f([\![\boldsymbol{v}_1]\!],\dots,[\![\boldsymbol{v}_k]\!])$ for $f(\boldsymbol{v}_1,\dots,\boldsymbol{v}_k) = \boldsymbol{v}_0$
  - $[\![\boldsymbol{v}]\!] \rightarrow [\![\boldsymbol{u}]\!]$ if $\exists\, \boldsymbol{w}$ s.t. $\boldsymbol{w} \odot \boldsymbol{v}, \boldsymbol{w} \odot \boldsymbol{u} \in D$

Soundness

- G_D : conjecture
- Proposition: $L(G_D) \subseteq L_0$
  - If $[\![\boldsymbol{v}]\!] \Rightarrow \boldsymbol{u}$ in G_D, then $L_0/\boldsymbol{v} = L_0/\boldsymbol{u}$
  - $[\![\boldsymbol{v}_0]\!] \rightarrow f([\![\boldsymbol{v}_1]\!],\ldots,[\![\boldsymbol{v}_k]\!])$ for $f(\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k) = \boldsymbol{v}_0$
  - $[\![\boldsymbol{v}]\!] \rightarrow [\![\boldsymbol{u}]\!]$ if $\exists\, \boldsymbol{w}$ s.t. $\boldsymbol{w} \odot \boldsymbol{v}, \boldsymbol{w} \odot \boldsymbol{u} \in D$

$L/\boldsymbol{v}_i = L/\boldsymbol{u}_i$ for $i = 1,\ldots,k$

$\Rightarrow L/f(\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k) = L/f(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k)$ for any non-erasing non-permuting f

Soundness

- G_D : conjecture
- Proposition: $L(G_D) \subseteq L_0$
  - If $[\![\boldsymbol{v}]\!] \Rightarrow \boldsymbol{u}$ in G_D, then $L_0/\boldsymbol{v} = L_0/\boldsymbol{u}$
  - $[\![\boldsymbol{v}_0]\!] \to f([\![\boldsymbol{v}_1]\!],\ldots,[\![\boldsymbol{v}_k]\!])$ for $f(\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k) = \boldsymbol{v}_0$
  - $[\![\boldsymbol{v}]\!] \to [\![\boldsymbol{u}]\!]$ if $\exists\, \boldsymbol{w}$ s.t. $\boldsymbol{w} \odot \boldsymbol{v}, \boldsymbol{w} \odot \boldsymbol{u} \in D$

$L/\boldsymbol{v}_i = L/\boldsymbol{u}_i$ for $i = 1,\ldots,k$
$\Rightarrow L/f(\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k) = L/f(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k)$ for any non-erasing non-permuting f

  - For $[\![v]\!] \in \mathcal{I}$ *and* $u \in L(G_D)$, v *and* u are congruent in L_0

Completeness

- G_D : conjecture, G_0 : target grammar s.t. $L_0 = L(G_0)$

Completeness

- G_D : conjecture, G_0 : target grammar s.t. $L_0 = L(G_0)$
- A: nonterminal of G_0

Completeness

- G_D : conjecture, G_0 : target grammar s.t. L_0 = L(G_0)
- A: nonterminal of G_0
- v_A : shortest in L(G_0, A)

Completeness

- G_D : conjecture, G_0 : target grammar s.t. $L_0 = \mathrm{L}(G_0)$
- A: nonterminal of G_0
- $\boldsymbol{v}_A$: shortest in $\mathrm{L}(G_0, A)$
- $\boldsymbol{w}_A$: shortest context for A, i.e., $\boldsymbol{w}_A \odot A$ is sentential

Completeness

- G_D : conjecture, G_0 : target grammar s.t. $L_0 = \mathrm{L}(G_0)$
- A: nonterminal of G_0
- $\boldsymbol{v}_A$: shortest in $\mathrm{L}(G_0, A)$
- $\boldsymbol{w}_A$: shortest context for A, i.e., $\boldsymbol{w}_A \odot A$ is sentential
- If $\{ \boldsymbol{w}_A \odot f(\boldsymbol{v}_B,\boldsymbol{v}_C,\boldsymbol{v}_E) \mid A \rightarrow f(B,C,E) \text{ in } G_0 \} \subseteq D$, then $L_0 \subseteq \mathrm{L}(G_D)$.

Completeness

- G_D : conjecture, G_0 : target grammar s.t. $L_0 = \mathrm{L}(G_0)$
- A: nonterminal of G_0
- $\boldsymbol{v}_A$: shortest in $\mathrm{L}(G_0, A)$
- $\boldsymbol{w}_A$: shortest context for A, i.e., $\boldsymbol{w}_A \odot A$ is sentential
- If $\{\ \boldsymbol{w}_A \odot f(\boldsymbol{v}_B,\boldsymbol{v}_C,\boldsymbol{v}_E) \mid A \rightarrow f(B,C,E) \text{ in } G_0\ \} \subseteq D$, then $L_0 \subseteq \mathrm{L}(G_D)$.
    - $A \rightarrow f(B,C,E)$ in G_0

Completeness

- G_D : conjecture, G_0 : target grammar s.t. $L_0 = \mathrm{L}(G_0)$
- A: nonterminal of G_0
- $\mathbf{v}_A$: shortest in $\mathrm{L}(G_0, A)$
- $\mathbf{w}_A$: shortest context for A, i.e., $\mathbf{w}_A \odot A$ is sentential
- If $\{ \mathbf{w}_A \odot f(\mathbf{v}_B,\mathbf{v}_C,\mathbf{v}_E) \mid A \to f(B,C,E) \text{ in } G_0 \} \subseteq D$, then $L_0 \subseteq \mathrm{L}(G_D)$.
  - $A \to f(B,C,E)$ in G_0
  - $\mathbf{w}_A \odot f(\mathbf{v}_B,\mathbf{v}_C,\mathbf{v}_E)$, $\mathbf{w}_A \odot \mathbf{v}_A \in D$

Completeness

- G_D : conjecture, G_0 : target grammar s.t. $L_0 = L(G_0)$
- A: nonterminal of G_0
- $\mathbf{v}_A$: shortest in $L(G_0, A)$
- $\mathbf{w}_A$: shortest context for A, i.e., $\mathbf{w}_A \odot A$ is sentential
- If $\{\, \mathbf{w}_A \odot f(\mathbf{v}_B, \mathbf{v}_C, \mathbf{v}_E) \mid A \to f(B,C,E) \text{ in } G_0 \,\} \subseteq D$, then $L_0 \subseteq L(G_D)$.
  - $A \to f(B,C,E)$ in G_0
  - $\mathbf{w}_A \odot f(\mathbf{v}_B, \mathbf{v}_C, \mathbf{v}_E)$, $\mathbf{w}_A \odot \mathbf{v}_A \in D$
  - $[\![\mathbf{v}_A]\!] \to [\![f(\mathbf{v}_B, \mathbf{v}_C, \mathbf{v}_E)]\!]$, $[\![f(\mathbf{v}_B, \mathbf{v}_C, \mathbf{v}_E)]\!] \to f([\![\mathbf{v}_B]\!], [\![\mathbf{v}_C]\!], [\![\mathbf{v}_E]\!])$ in G_D

Completeness

- G_D : conjecture, G_0 : target grammar s.t. $L_0 = \mathrm{L}(G_0)$
- A: nonterminal of G_0
- $\boldsymbol{v}_A$: shortest in $\mathrm{L}(G_0, A)$
- $\boldsymbol{w}_A$: shortest context for A, i.e., $\boldsymbol{w}_A \odot A$ is sentential
- If $\{ \boldsymbol{w}_A \odot f(\boldsymbol{v}_B,\boldsymbol{v}_C,\boldsymbol{v}_E) \mid A \to f(B,C,E) \text{ in } G_0 \} \subseteq D$, then $L_0 \subseteq \mathrm{L}(G_D)$.
  - $A \to f(B,C,E)$ in G_0
  - $\boldsymbol{w}_A \odot f(\boldsymbol{v}_B,\boldsymbol{v}_C,\boldsymbol{v}_E)$, $\boldsymbol{w}_A \odot \boldsymbol{v}_A \in D$
  - $[\![\boldsymbol{v}_A]\!] \to [\![f(\boldsymbol{v}_B,\boldsymbol{v}_C,\boldsymbol{v}_E)]\!]$, $[\![f(\boldsymbol{v}_B,\boldsymbol{v}_C,\boldsymbol{v}_E)]\!] \to f([\![\boldsymbol{v}_B]\!],[\![\boldsymbol{v}_C]\!],[\![\boldsymbol{v}_E]\!])$ in G_D
- $[\![v_S]\!]$ is an initial symbol for $v_S \in D$

- $(m+1)$D-substitutability is stronger than mD-substitutability

- (m+1)D-substitutability is stronger than mD-substitutability
- $\{ a^n b c^n d e^n \mid n \geqq 0 \}$ is 2D-substitutable

- $(m+1)$D-substitutability is stronger than mD-substitutability
- $\{ a^n b c^n d e^n \mid n \geqq 0 \}$ is 2D-substitutable
- $\{ a^n c^n e^n \mid n \geqq 0 \}$ is not 2D-substitutable

- (m+1)D-substitutability is stronger than mD-substitutability
- $\{ a^n b c^n d e^n \mid n \geqq 0 \}$ is 2D-substitutable
- $\{ a^n c^n e^n \mid n \geqq 0 \}$ is not 2D-substitutable
  - $\{ a^n c^n \mid n \geqq 0 \}$ is not (1D-)substitutable

- (m+1)D-substitutability is stronger than mD-substitutability
- $\{ a^n b c^n d e^n \mid n \geqq 0 \}$ is 2D-substitutable
- $\{ a^n c^n e^n \mid n \geqq 0 \}$ is not 2D-substitutable
    - $\{ a^n c^n \mid n \geqq 0 \}$ is not (1D-)substitutable
- $L = \{ aaabaaa \}$ is not 2D-substitutable

- $(m+1)$D-substitutability is stronger than mD-substitutability
- $\{ a^n b c^n d e^n \mid n \geqq 0 \}$ is 2D-substitutable
- $\{ a^n\ c^n\ e^n \mid n \geqq 0 \}$ is not 2D-substitutable
  - $\{ a^n\ c^n \mid n \geqq 0 \}$ is not (1D-)substitutable
- $L = \{ aaabaaa \}$ is not 2D-substitutable
  - $(\mathrm{a\_a\_a}) \in L/(aab, a) \cap L/(a, baa)$

- $(m+1)$D-substitutability is stronger than mD-substitutability
- $\{ a^n b c^n d e^n \mid n \geqq 0 \}$ is 2D-substitutable
- $\{ a^n\ c^n\ e^n \mid n \geqq 0 \}$ is not 2D-substitutable
    - $\{ a^n\ c^n \mid n \geqq 0 \}$ is not (1D-)substitutable
- $L = \{ aaabaaa \}$ is not 2D-substitutable
    - (a_a_a) $\in L/(aab, a) \cap L/(a, baa)$
    - (a_aa_ε) $\in L/(aab, a) - L/(a, baa)$

p-Finite Kernel Property

- A set of multiwords $\boldsymbol{V}_A$ is a ***p*-kernel** of L(G, A) (or of $A \in N$) iff
  iff L(G)/$\boldsymbol{V}_A$ = L(G)/L(A)
  (i.e., $\boldsymbol{w} \odot \boldsymbol{V}_A \in L(G) \Rightarrow \boldsymbol{w} \odot \mathrm{L}(A) \subseteq L(G)$ for any multi-context $\boldsymbol{w}$)

- MCFG G has the ***p*-Finite Kernel Property**
  iff every nonterminal admits a *p*-kernel

p-Finite Kernel Property

- A set of multiwords $\boldsymbol{V}_A$ is a ***p*-kernel** of L(G, A) (or of $A \in N$) iff
  iff L(G)/$\boldsymbol{V}_A$ = L(G)/L(A)
  (i.e., $\boldsymbol{w} \odot \boldsymbol{V}_A \in L(G) \;\Rightarrow\; \boldsymbol{w} \odot \mathrm{L}(A) \subseteq L(G)$ for any multi-context $\boldsymbol{w}$)

- MCFG G has the ***p*-Finite Kernel Property**
  iff every nonterminal admits a p-kernel

- L has the p-FKP if it is generated by a grammar with the p-FKP

p-Finite Kernel Property

- A set of multiwords $\boldsymbol{V}_A$ is a ***p*-kernel** of L(G, A) (or of $A \in N$)   iff
  iff L(G)/$\boldsymbol{V}_A$ = L(G)/L(A)
  (i.e., $\boldsymbol{w} \odot \boldsymbol{V}_A \in L(G) \Rightarrow \boldsymbol{w} \odot \mathrm{L}(A) \subseteq L(G)$ for any multi-context $\boldsymbol{w}$)

- MCFG G has the ***p*-Finite Kernel Property**
  iff every nonterminal admits a p-kernel

- L has the p-FKP if it is generated by a grammar with the p-FKP

- m-MCFL(n) with the p-FKP covers CFLs with the p-FKP

p-Finite Kernel Property

- A set of multiwords $\boldsymbol{V}_A$ is a ***p*-kernel** of L(G, A) (or of $A \in N$) iff
  iff L(G)/$\boldsymbol{V}_A$ = L(G)/L(A)
  (i.e., $\boldsymbol{w} \odot \boldsymbol{V}_A \in L(G) \;\Rightarrow\; \boldsymbol{w} \odot \mathrm{L}(A) \subseteq L(G)$ for any multi-context $\boldsymbol{w}$)

- MCFG G has the ***p*-Finite Kernel Property**
  iff every nonterminal admits a *p*-kernel

- *L* has the *p*-FKP if it is generated by a grammar with the *p*-FKP

- *m*-MCFL*(n)* with the *p*-FKP covers CFLs with the *p*-FKP

- Identification in the limit from positive data and membership queries

Learner's Conjecture

- $K_i \subseteq \mathrm{Sub}_i(D)$ for $i = 1,\dots,n$ for D positive data
- $X_i = \mathrm{Con}_i(D) = \{ \boldsymbol{w} \mid \boldsymbol{w} \odot \boldsymbol{v} \in D$ for some $\boldsymbol{v} \in K_i \}$
- $G_{K,X}$: conjecture
  - $N_i = \{ [\![\boldsymbol{V}]\!] \mid \boldsymbol{V} \subseteq K_i, |\boldsymbol{V}| \leqq p \}$
  - ★ $[\![\boldsymbol{V}]\!] \Rightarrow \boldsymbol{u}$ for $L_0/\boldsymbol{V} \subseteq L_0/\boldsymbol{u}$
  - Initial Symbols: $\{ [\![V]\!] \in N_1 \mid V \subseteq L_0 \}$
  - Rules
    - $[\![\boldsymbol{V}]\!] \rightarrow f([\![\boldsymbol{V}_1]\!],\dots,[\![\boldsymbol{V}_k]\!])$
      if $L_0/\boldsymbol{V} \cap X \subseteq L_0/f(\boldsymbol{V}_1,\dots,\boldsymbol{V}_k) \cap X$
      and every substring in $\boldsymbol{t}_i$ is from an element of K_1

Learner's Conjecture

- CFG Learning
  - $K \subseteq \{ V \mid V \subseteq \mathrm{Sub}_1(D) \text{ and } |V| \leqq p \}$,
    $X = \mathrm{Con}_1(D)$,
    $N = \{ [\![V]\!] \mid V \in K \}$
  - Initial Symbols: $\{ [\![V]\!] \in N \mid V \subseteq D \}$
  - Rules $[\![V]\!] \to [\![V_1]\!][\![V_2]\!]$ if $L/V \cap X \subseteq L/V_1V_2 \cap X$

    $[\![V]\!] \to a$ if $L/V \cap X \subseteq L/a \cap X$

Learner's Conjecture

- CFG Learning
  - $K \subseteq \{ V \mid V \subseteq \mathrm{Sub}_1(D) \text{ and } |V| \leqq p \}$,
    $X = \mathrm{Con}_1(D)$,
    $N = \{ [\![V]\!] \mid V \in K \}$
  - Initial Symbols: $\{ [\![V]\!] \in N \mid V \subseteq D \}$
  - Rules $[\![V]\!] \to [\![V_1]\!][\![V_2]\!]$ if $L/V \cap X \subseteq L/V_1V_2 \cap X$

    $[\![V]\!] \to a$ if $L/V \cap X \subseteq L/a \cap X$
- MCFG Learning
  - $N_k = \{ [\![V]\!] \mid V \in K_k \}$ for k $\leqq m$, where
    $K_k = \{ \mathbf{V} \mid \mathbf{V} \subseteq \mathrm{Sub}_k(D) \text{ and } |\mathbf{V}| \leqq p \}$ for k $\leqq m$,
    $X = \mathrm{Con}(D)$,
  - Initial Symbols: $\{ [\![V]\!] \in N_1 \mid V \subseteq D \}$
  - Rules: $[\![\mathbf{V}_0]\!] \to f([\![\mathbf{V}_1]\!],\ldots,[\![\mathbf{V}_k]\!])$

    if $L/\mathbf{V}_0 \cap X \subseteq L/f(\mathbf{V}_1,\ldots,\mathbf{V}_k) \cap X$ with $k \leqq n$,

    $f(\mathbf{v}_1,\ldots,\mathbf{v}_k) \in K$ for some $\mathbf{v}_1,\ldots,\mathbf{v}_k \in K$

Learning m-MCFG(n) with p-FKP

Let D := K := X := ∅; G := vacuous grammar;
For i = 1,2,3,...
  let $D := \{w_1, w_2, \ldots, w_i\}$;
  If $D \not\subseteq L(G)$
    then let K_i := $\mathrm{Sub}_i(D)$ for $i = 1,\ldots,m$;
  End if
  let X_i := $\mathrm{Con}_i(D)$ for $i = 1,\ldots,m$;
  update G with $K = \bigcup_i K_i$ and $X = \bigcup_i X_i$
End for

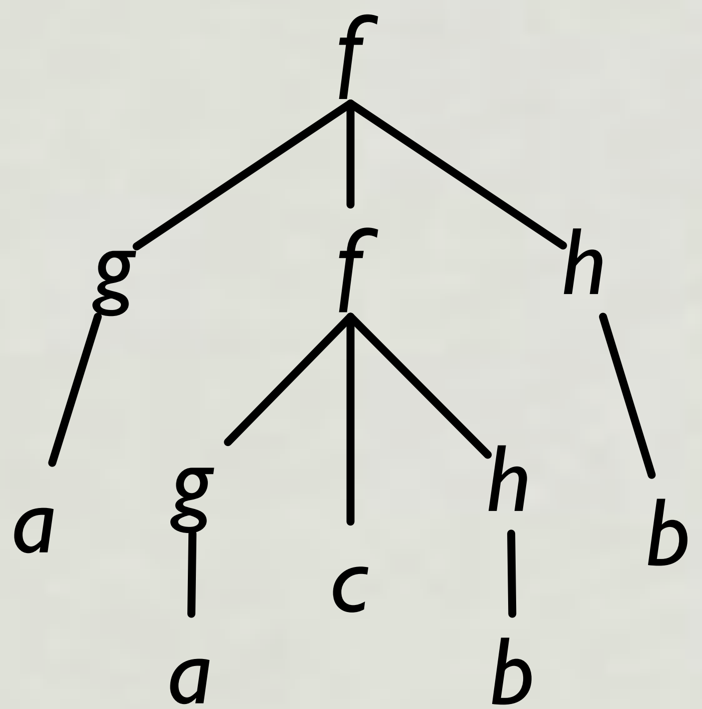
Theorem

- *m*-MCFL*(n)* with the *p*-FKP is identifiable in the limit from positive data & membership queries
- Polynomial-time update (when *m,n,p* are fixed)
- Polynomially many examples are enough for convergence w.r.t. the size of the grammar to be learnt

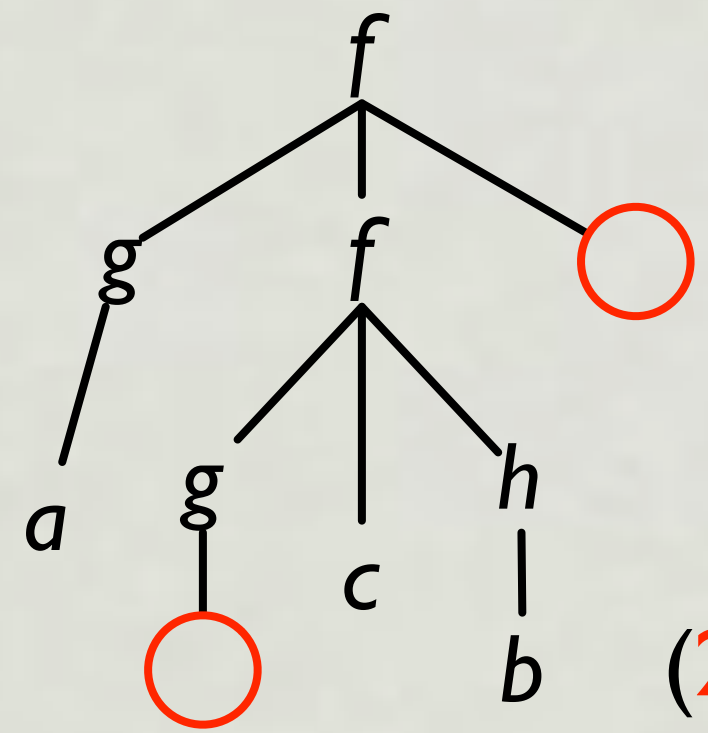
Other Related Formalisms

Trees and Stubs

Tree (0-stub):



m-Stub (= m tree context):
tree with m "open leaves"

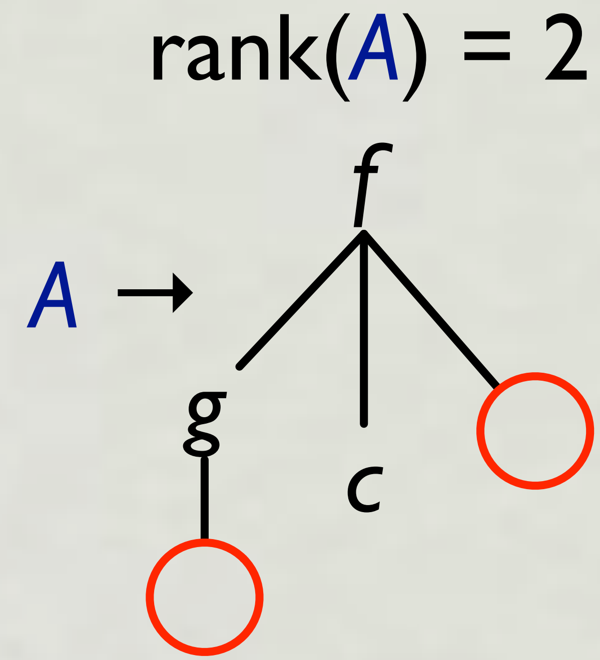


(2-Stub)

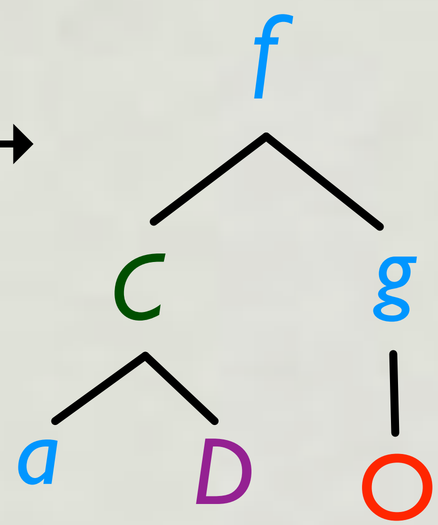
rank(a) = rank(b) = rank(c) = 0, rank(g) = rank(h) = 1, rank(f) = 3

r-Simple Context-Free Tree Grammars

- $G = (N, \Sigma, P, I)$
  - N, Σ: ranked nonterminal/terminal symbols
  - Rank is at most r
  - $P \subseteq \bigcup_k N_k \times (k\text{-Stubs})$: production rules
  - $I \subseteq N_0$: initial symbols of rank 0
- If $A \to s[\circ,\circ,\circ]$ is in P, then $t[A[t_1,t_2,t_3]] \Rightarrow t[s[t_1,t_2,t_3]]$
- $L(G) = \{ t \mid S \Rightarrow t$ for some $S \in I$ and t is a tree over $\Sigma \}$
- every 1-SCFTG can be identified with a CFG

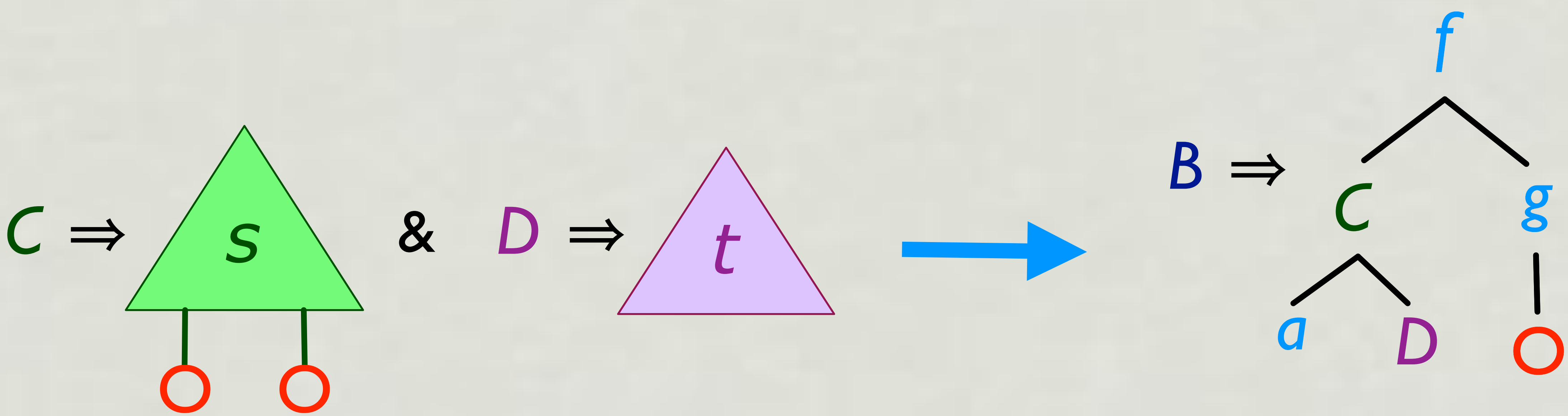


Simple Context-Free Tree Grammars

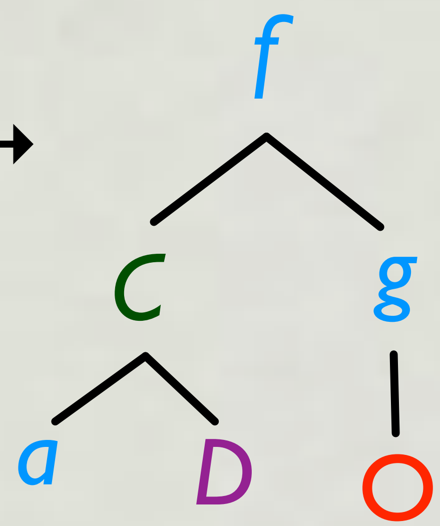
- $B \rightarrow$ 

★ *a,f,g* : terminal symbol

rank(*B*) = 1
rank(*C*) = 2
rank(*D*) = 0

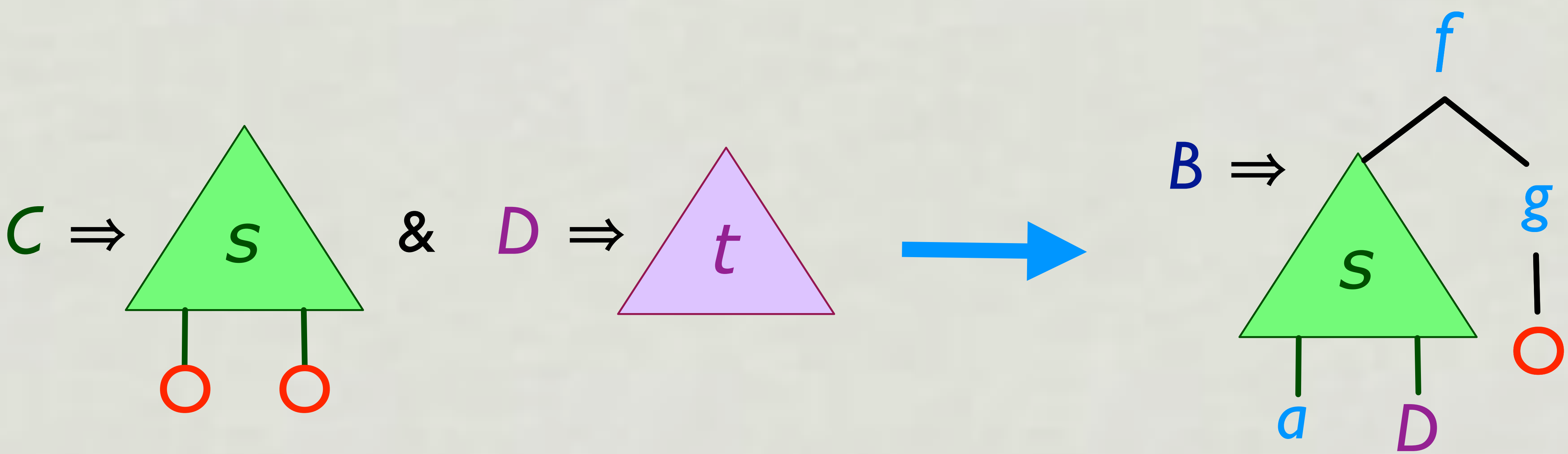


Simple Context-Free Tree Grammars

- $B \rightarrow$ 

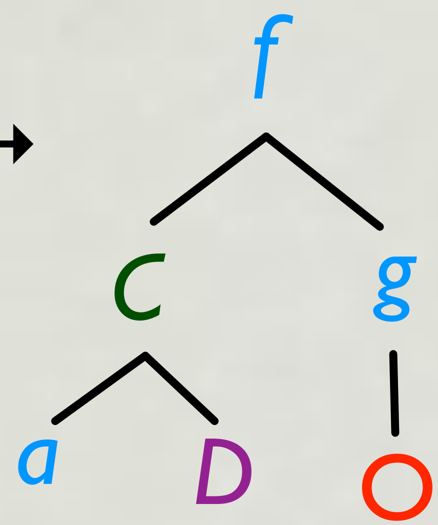
★ *a,f,g* : terminal symbol

rank(*B*) = 1
rank(*C*) = 2
rank(*D*) = 0



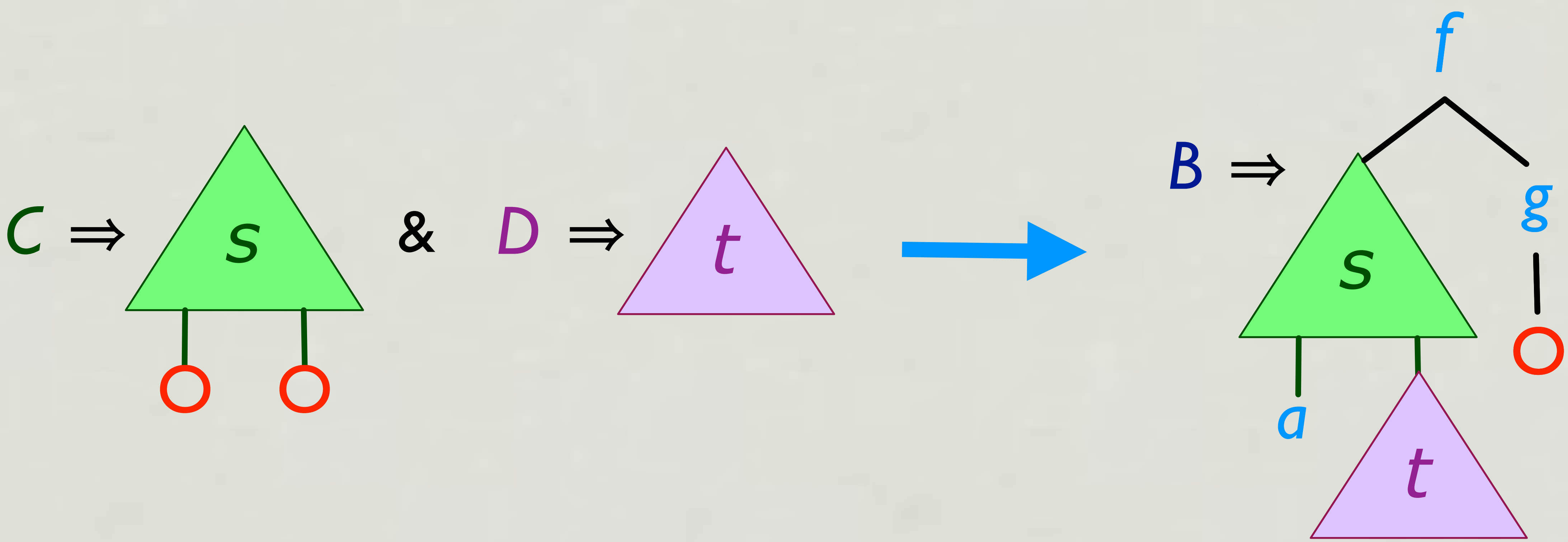
Simple Context-Free Tree Grammars

- $B \rightarrow$



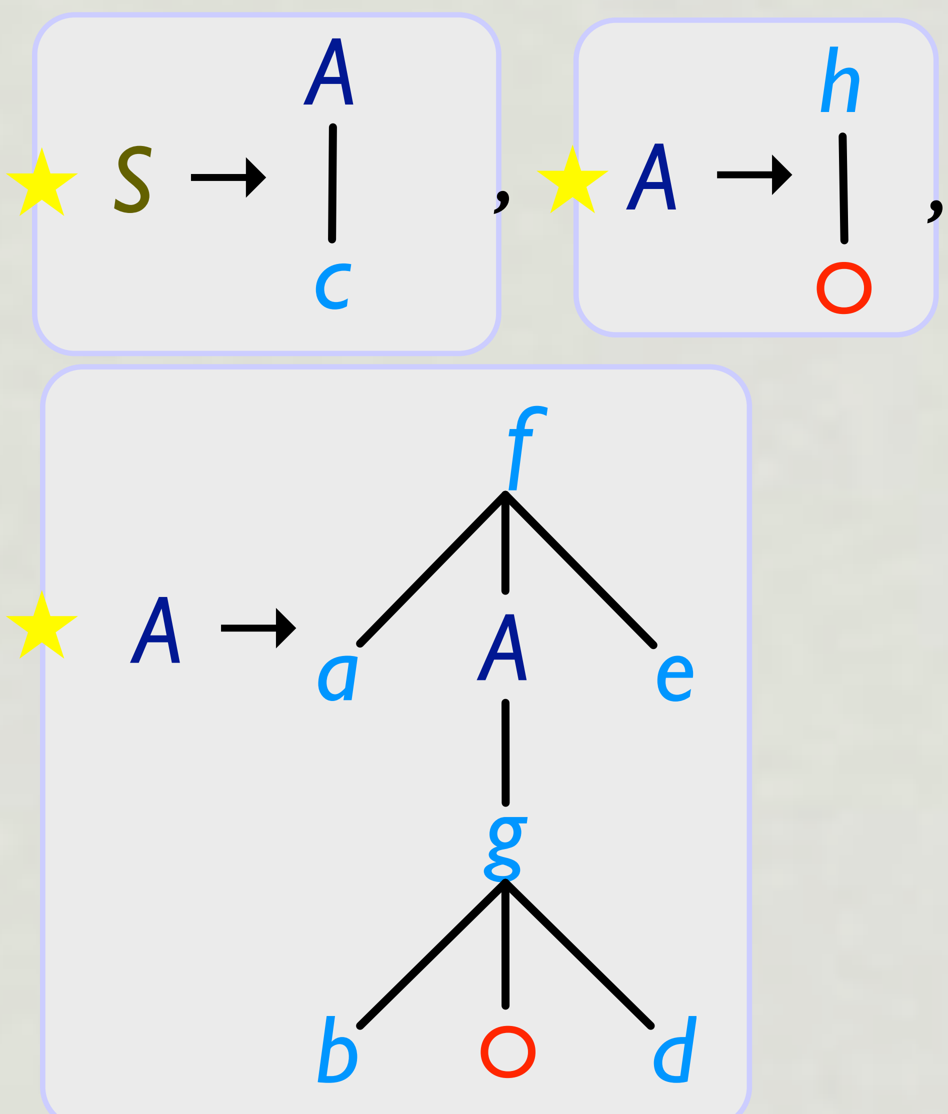
★ *a,f,g* : terminal symbol

rank(*B*) = 1
rank(*C*) = 2
rank(*D*) = 0



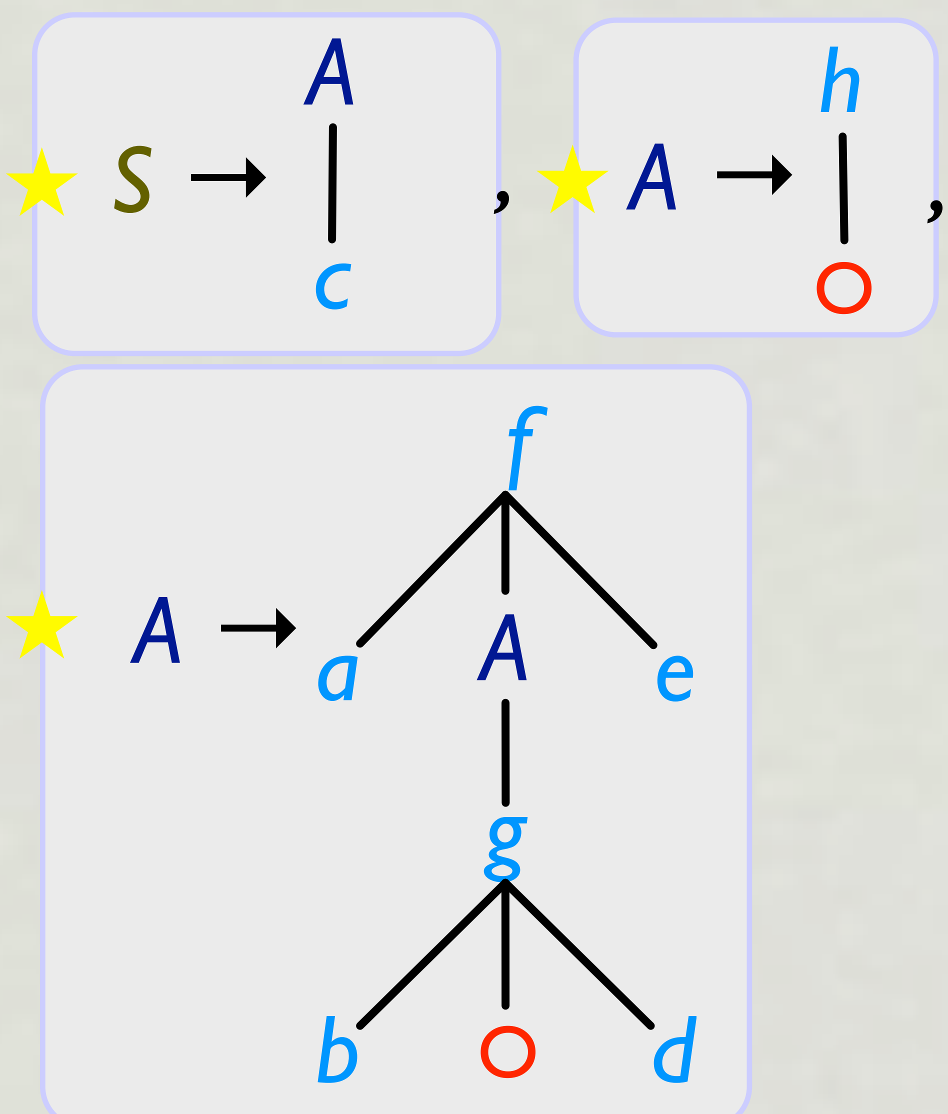
Example

- $N_0 = \{ S \}$, $N_1 = \{ A \}$
- $\Sigma_0 = \{ a,b,c,d,e \}$, $\Sigma_1 = \{ h \}$, $\Sigma_3 = \{ f,g \}$



Example

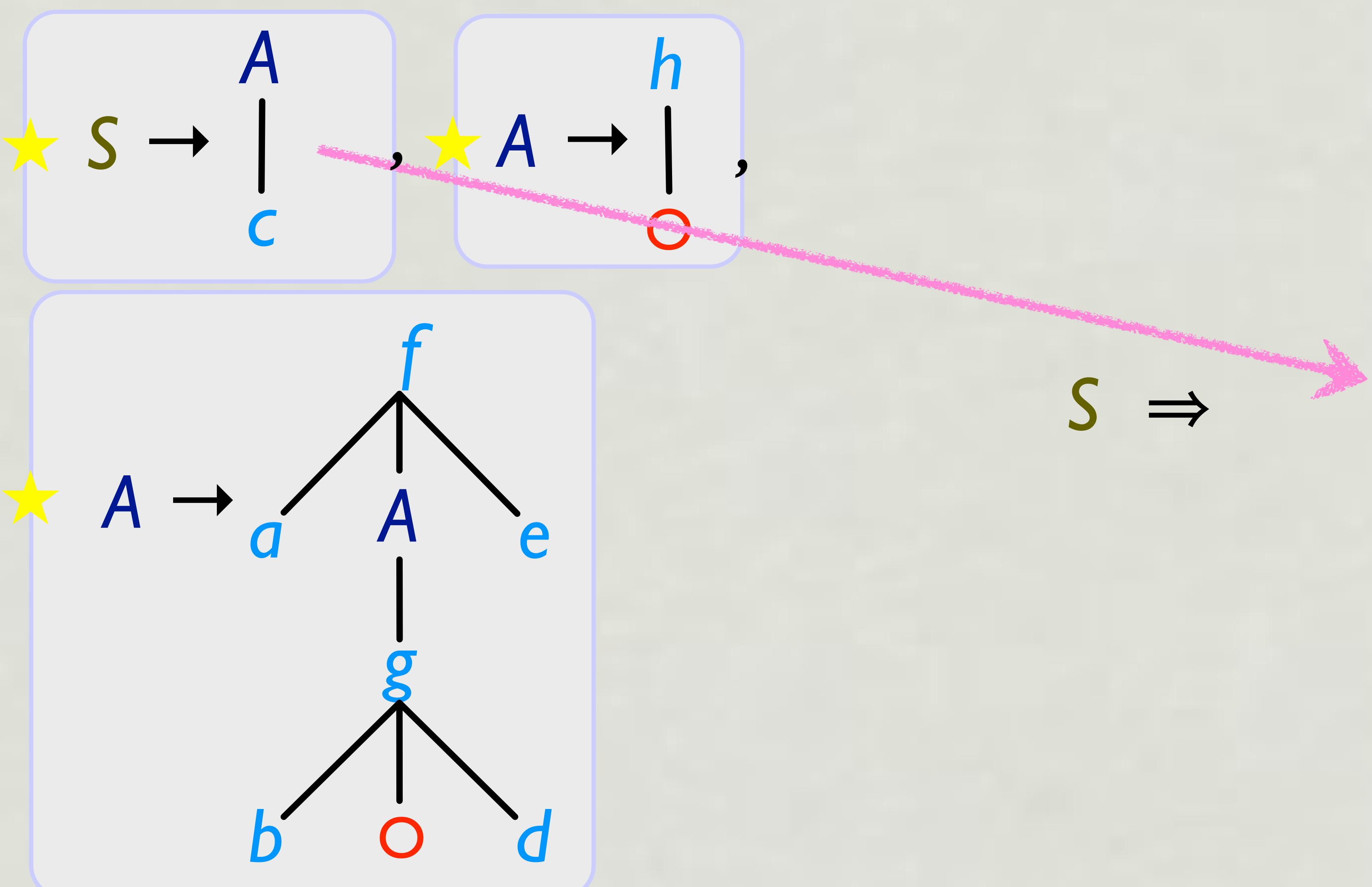
- $N_0 = \{ S \}$, $N_1 = \{ A \}$
- $\Sigma_0 = \{ a,b,c,d,e \}$, $\Sigma_1 = \{ h \}$, $\Sigma_3 = \{ f,g \}$



$S \Rightarrow$

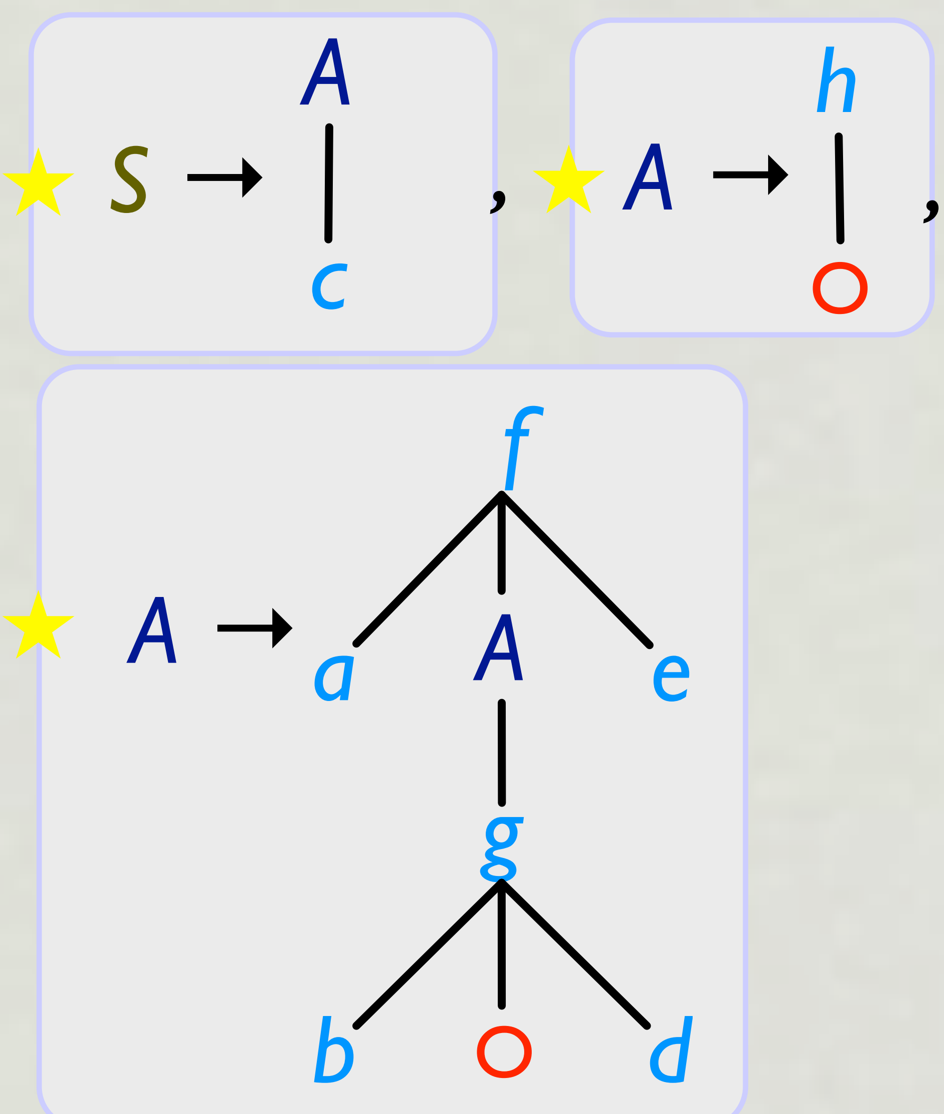
Example

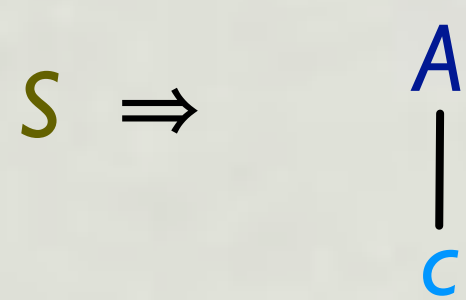
- $N_0 = \{ S \}$, $N_1 = \{ A \}$
- $\Sigma_0 = \{ a,b,c,d,e \}$, $\Sigma_1 = \{ h \}$, $\Sigma_3 = \{ f,g \}$



Example

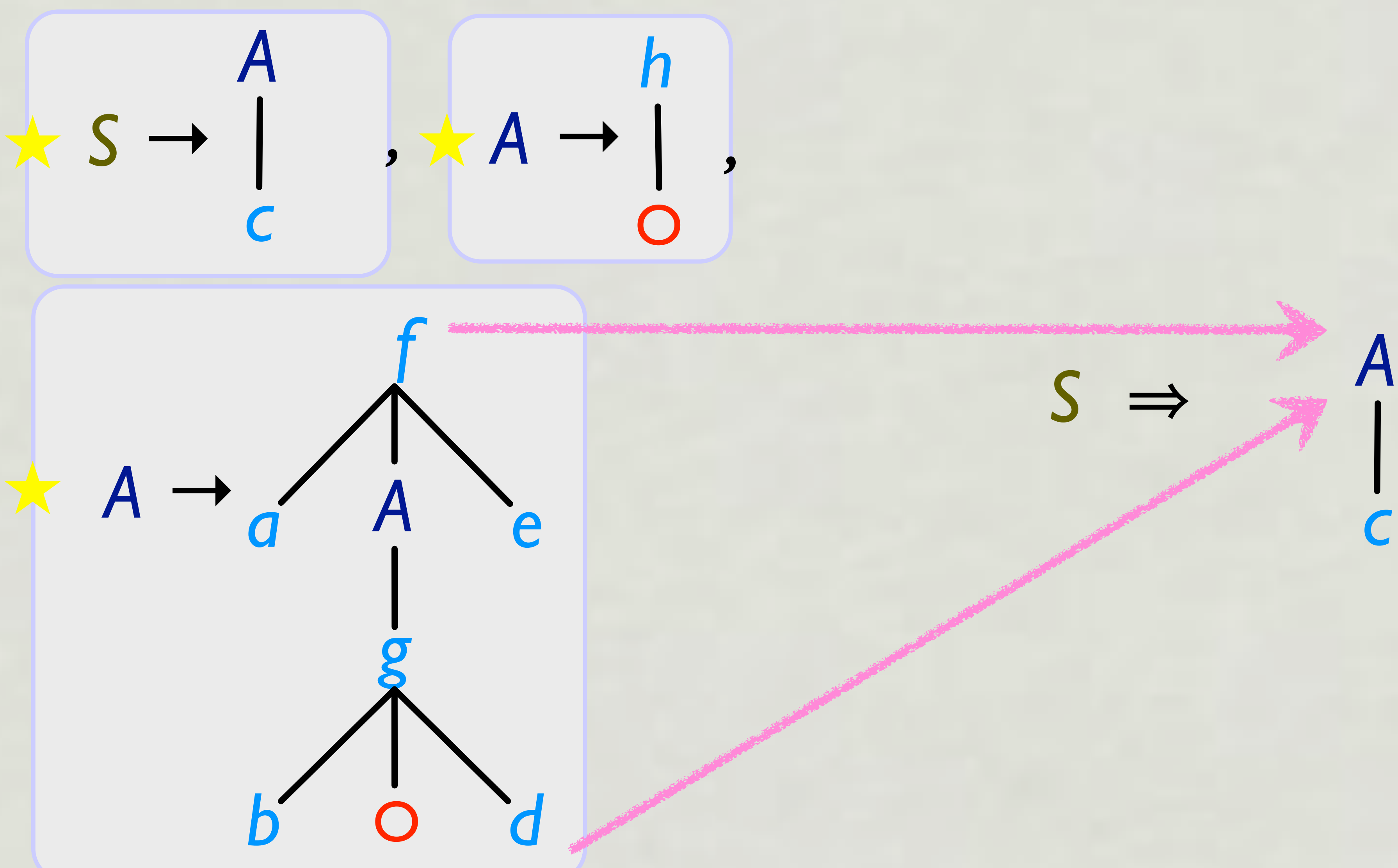
- $N_0 = \{ S \}$, $N_1 = \{ A \}$
- $\Sigma_0 = \{ a,b,c,d,e \}$, $\Sigma_1 = \{ h \}$, $\Sigma_3 = \{ f,g \}$





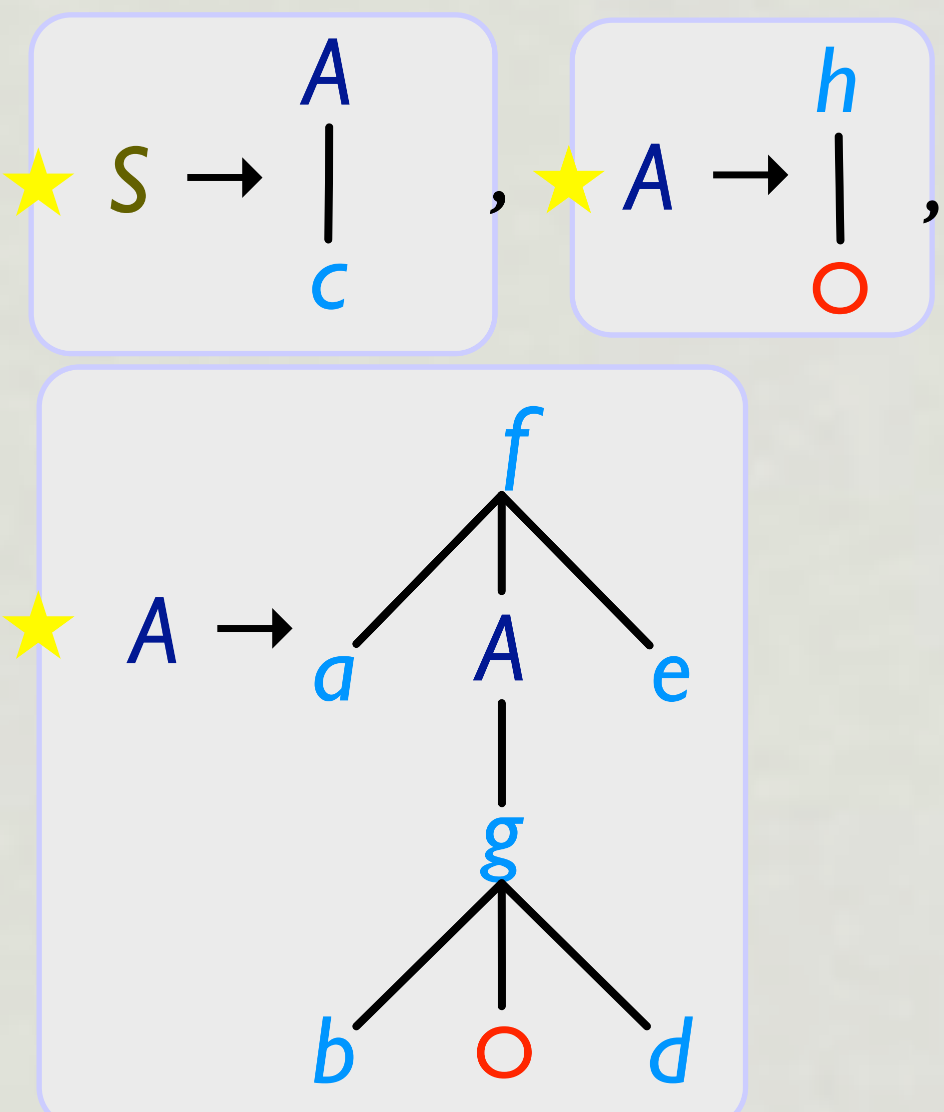
Example

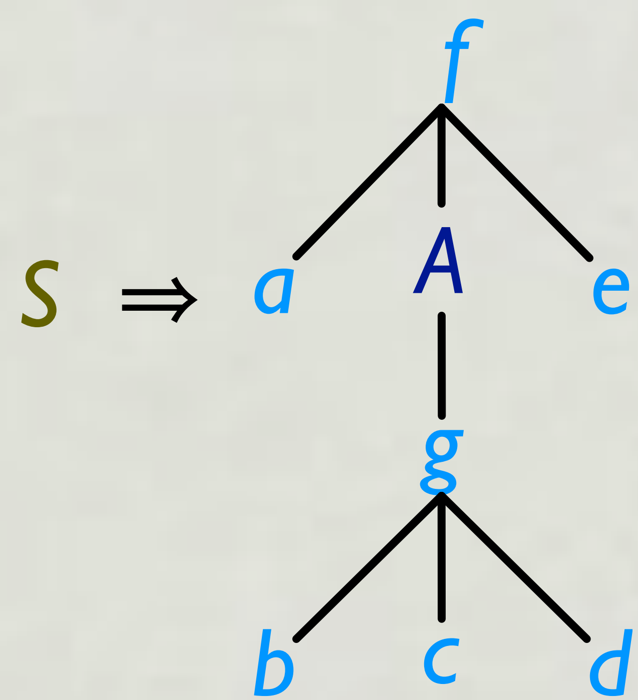
- $N_0 = \{ S \},\ N_1 = \{ A \}$
- $\Sigma_0 = \{ a,b,c,d,e \},\ \Sigma_1 = \{ h \},\ \Sigma_3 = \{ f,g \}$



Example

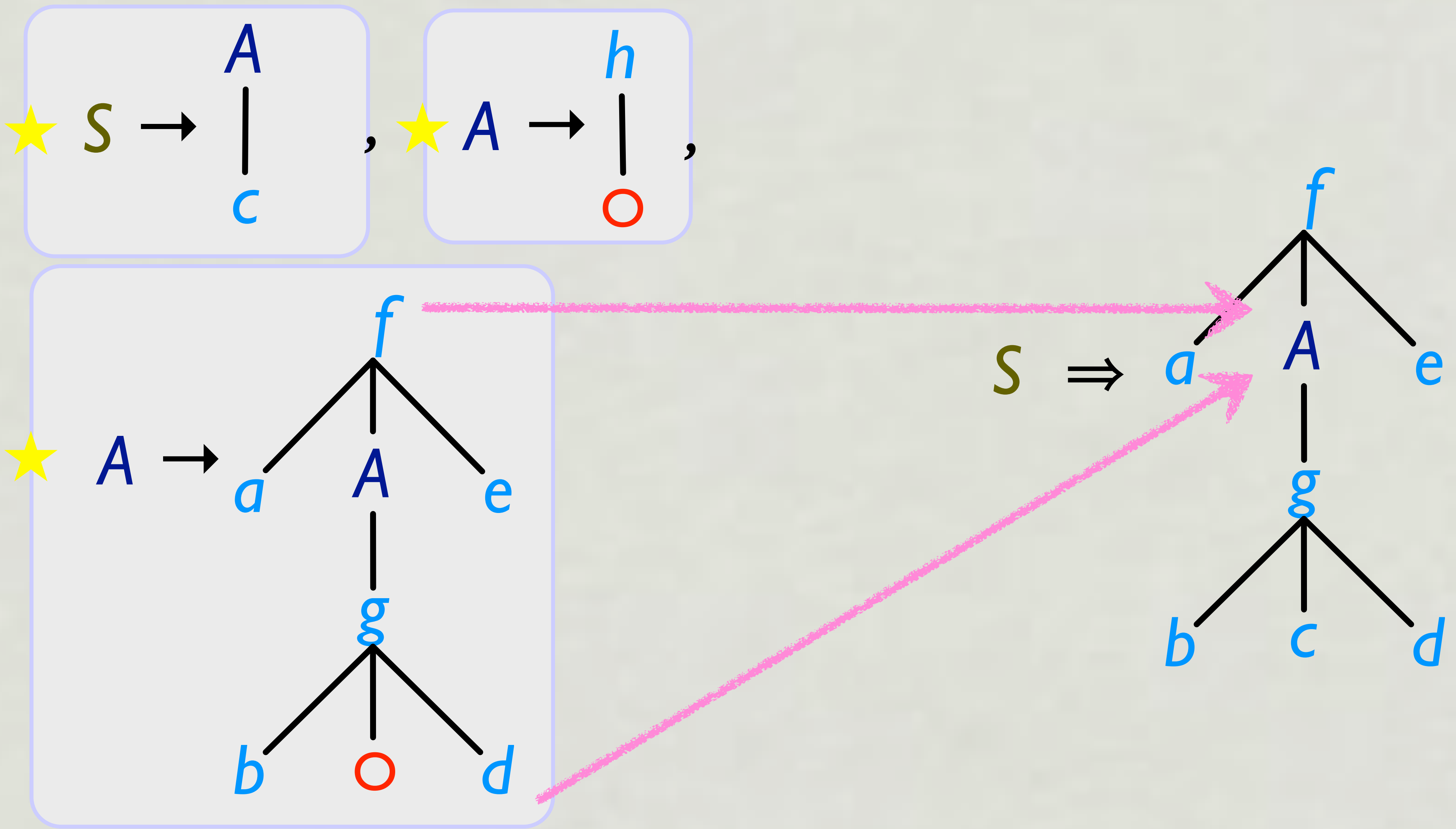
- $N_0 = \{ S \}$, $N_1 = \{ A \}$
- $\Sigma_0 = \{ a,b,c,d,e \}$, $\Sigma_1 = \{ h \}$, $\Sigma_3 = \{ f,g \}$





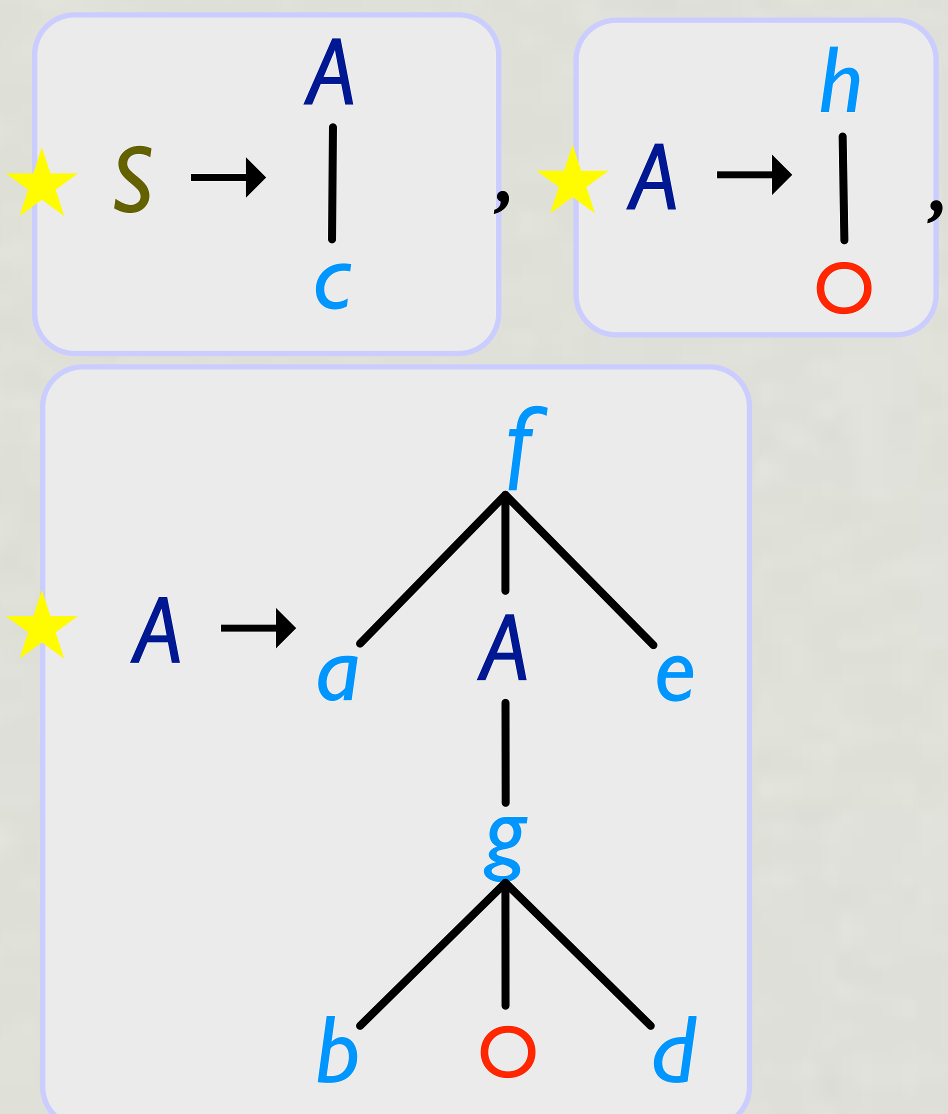
Example

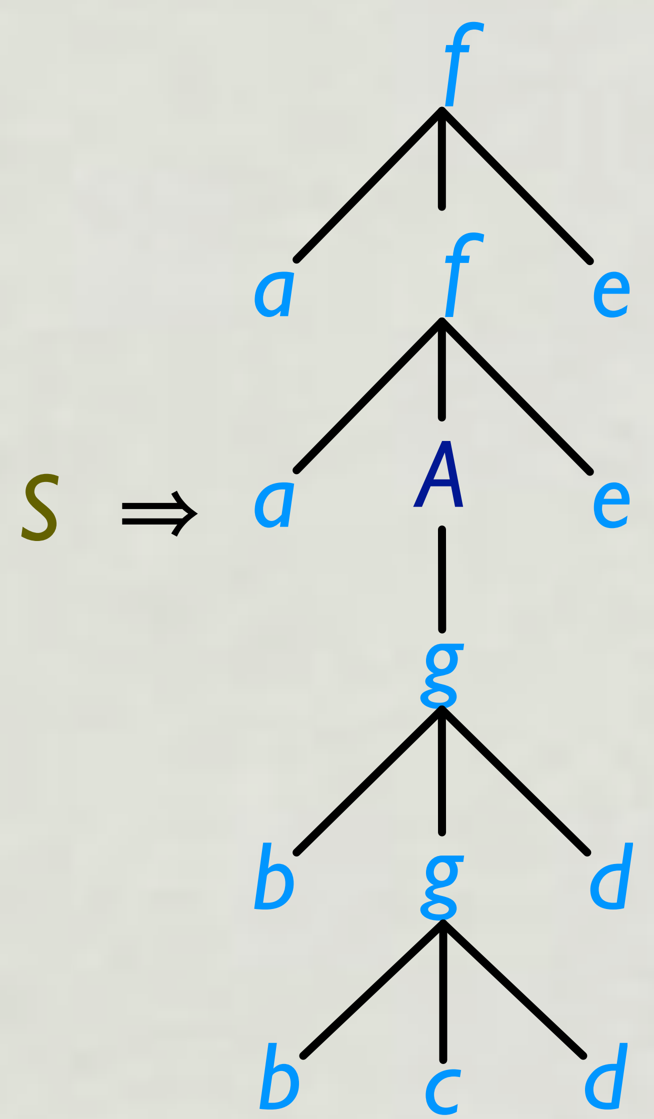
- $N_0 = \{ S \}$, $N_1 = \{ A \}$
- $\Sigma_0 = \{ a,b,c,d,e \}$, $\Sigma_1 = \{ h \}$, $\Sigma_3 = \{ f,g \}$



Example

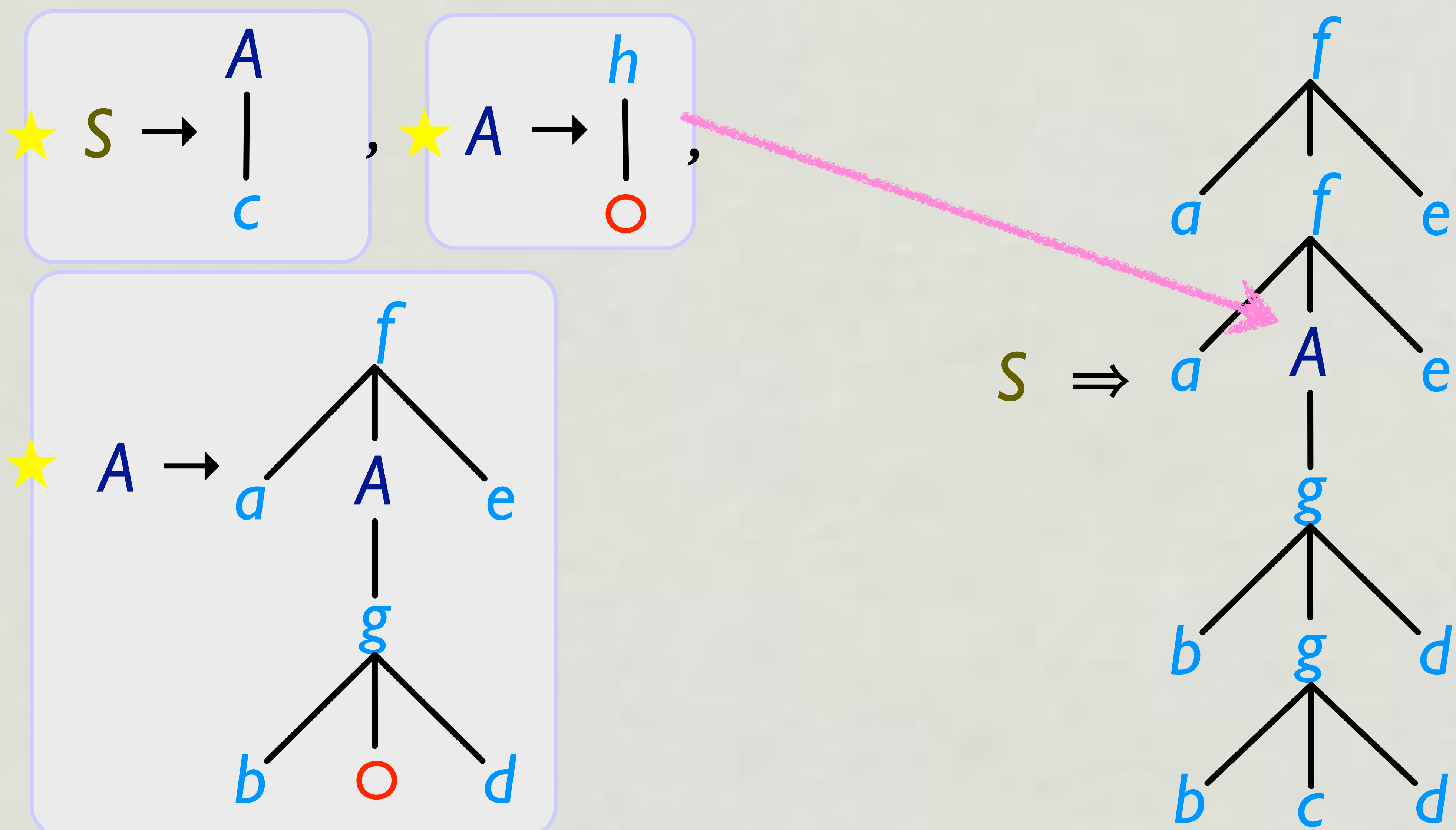
- $N_0 = \{ S \}$, $N_1 = \{ A \}$
- $\Sigma_0 = \{ a,b,c,d,e \}$, $\Sigma_1 = \{ h \}$, $\Sigma_3 = \{ f,g \}$





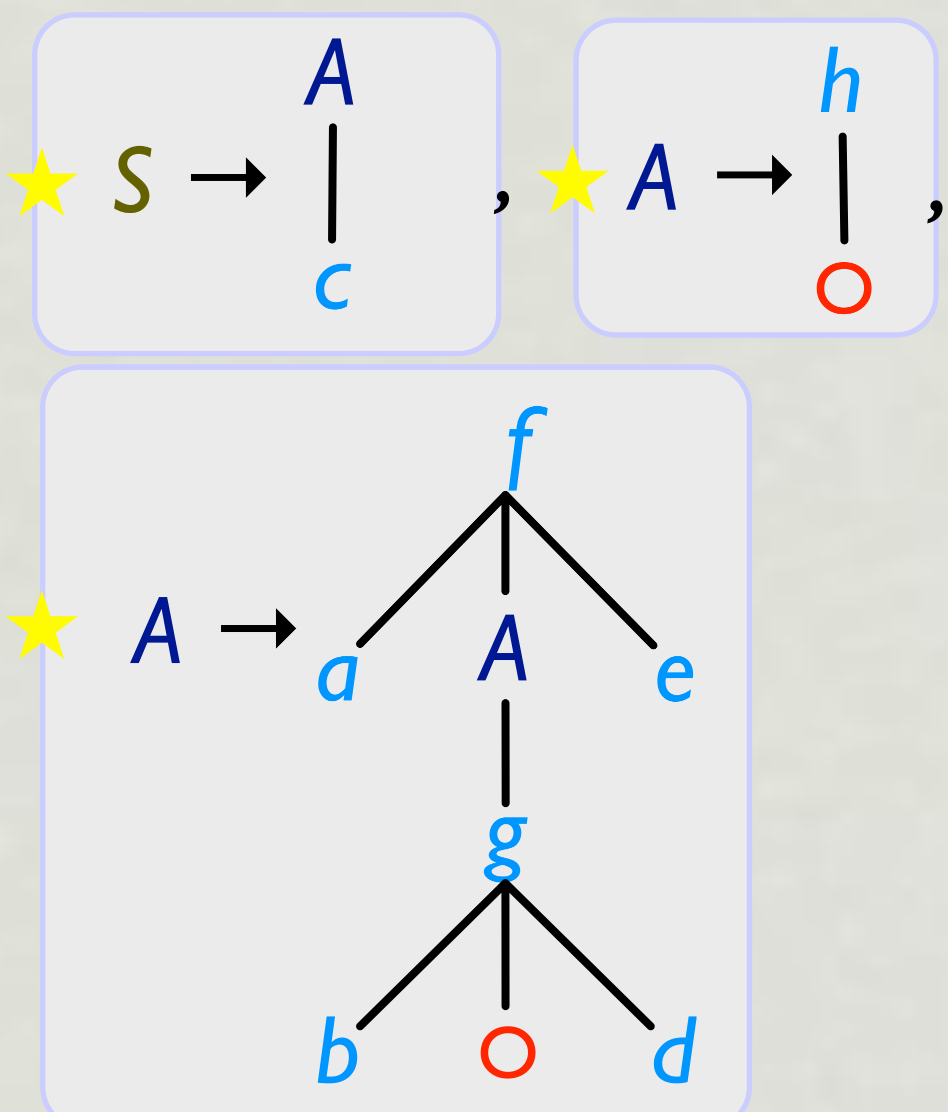
Example

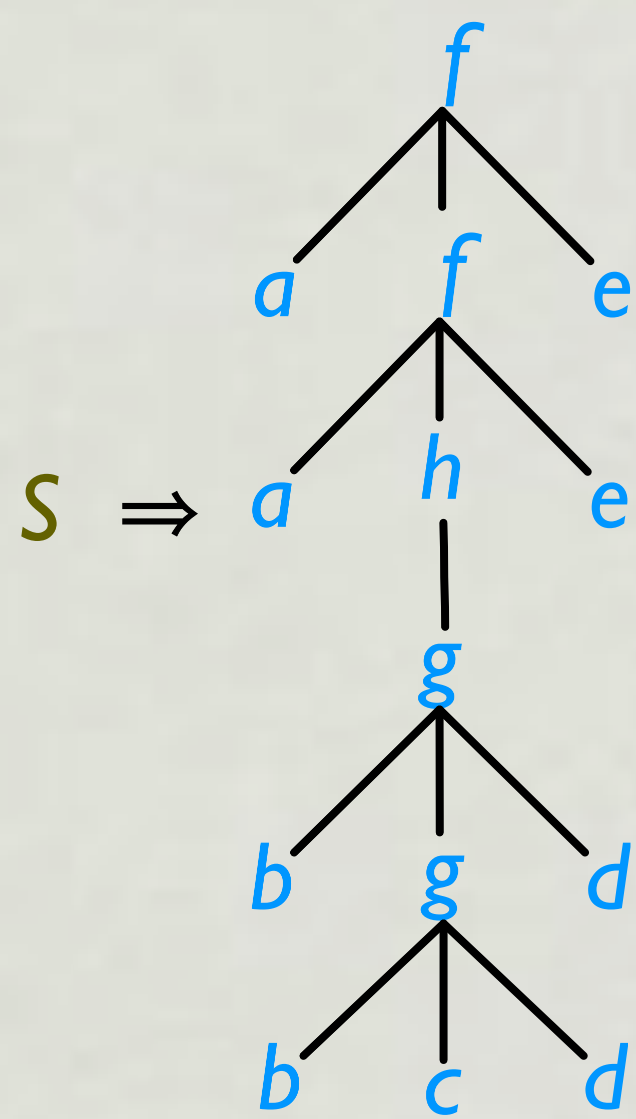
- $N_0 = \{ S \}$, $N_1 = \{ A \}$
- $\Sigma_0 = \{ a,b,c,d,e \}$, $\Sigma_1 = \{ h \}$, $\Sigma_3 = \{ f,g \}$



Example

- $N_0 = \{ S \}$, $N_1 = \{ A \}$
- $\Sigma_0 = \{ a,b,c,d,e \}$, $\Sigma_1 = \{ h \}$, $\Sigma_3 = \{ f,g \}$





Substructure/Context Decomposition

Simple Context-Free Tree Grammars

CFGs

- $S \Rightarrow u_1 A u_2$
- $A \Rightarrow v$
- $u_1 v u_2 \in L(G)$

(u_1, u_2): **context**

v : **substring**

Substructure/Context Decomposition

CFGs

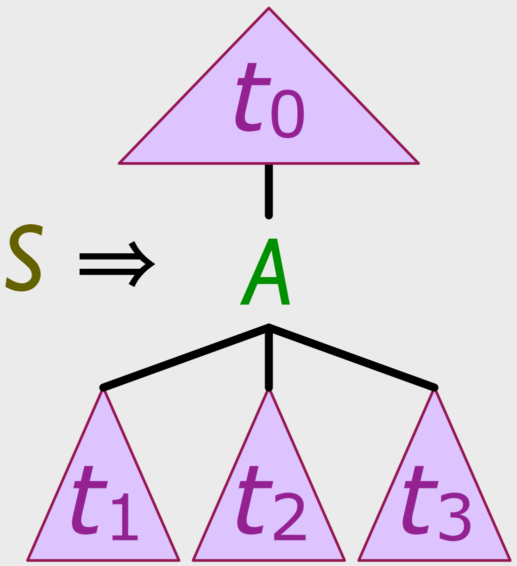
- $S \Rightarrow u_1 A u_2$
- $A \Rightarrow v$
- $u_1 v u_2 \in L(G)$

(u_1, u_2): **context**

v : **substring**

Simple Context-Free Tree Grammars

- $S \Rightarrow$



Substructure/Context Decomposition

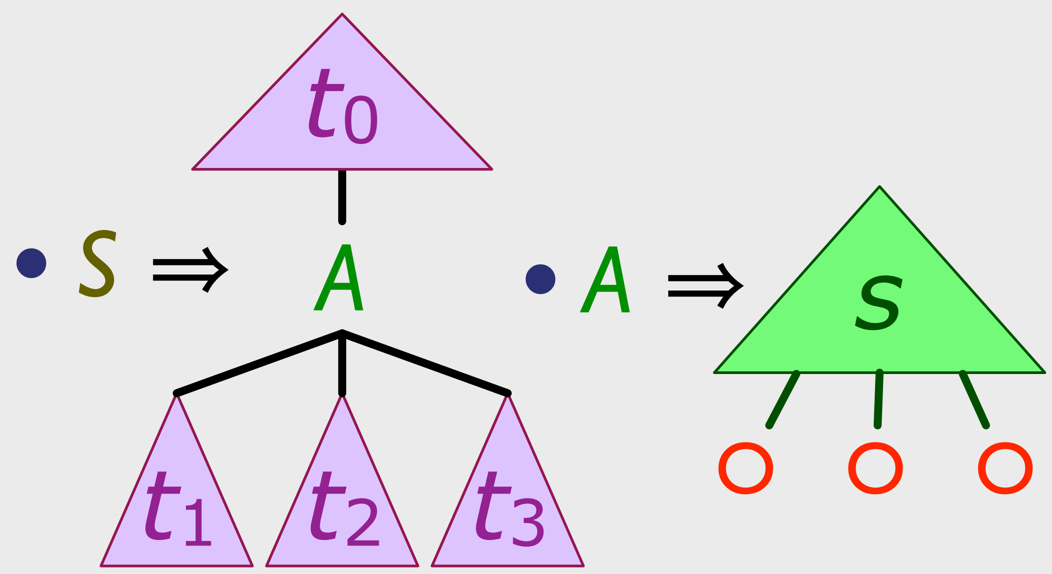
CFGs

- $S \Rightarrow u_1 A u_2$
- $A \Rightarrow v$
- $u_1 v u_2 \in L(G)$

(u_1, u_2): **context**

v : **substring**

Simple Context-Free Tree Grammars



Substructure/Context Decomposition

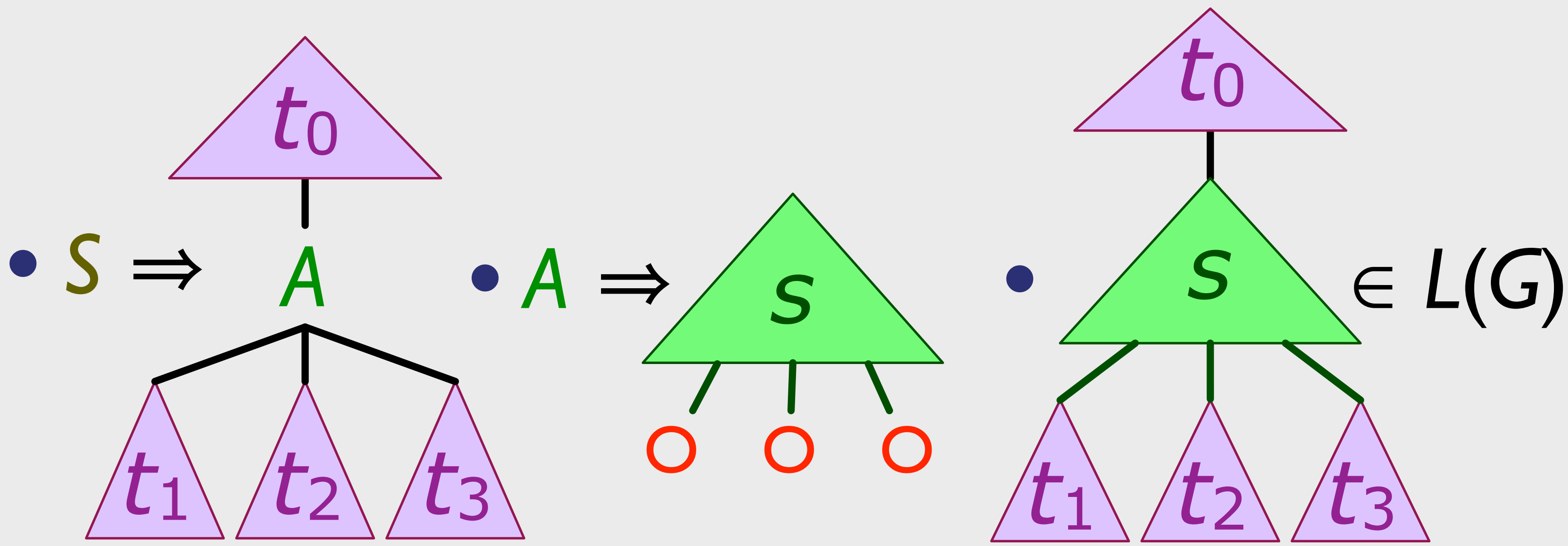
CFGs

- $S \Rightarrow u_1 A u_2$
- $A \Rightarrow v$
- $u_1 v u_2 \in L(G)$

(u_1, u_2): **context**

v : **substring**

Simple Context-Free Tree Grammars



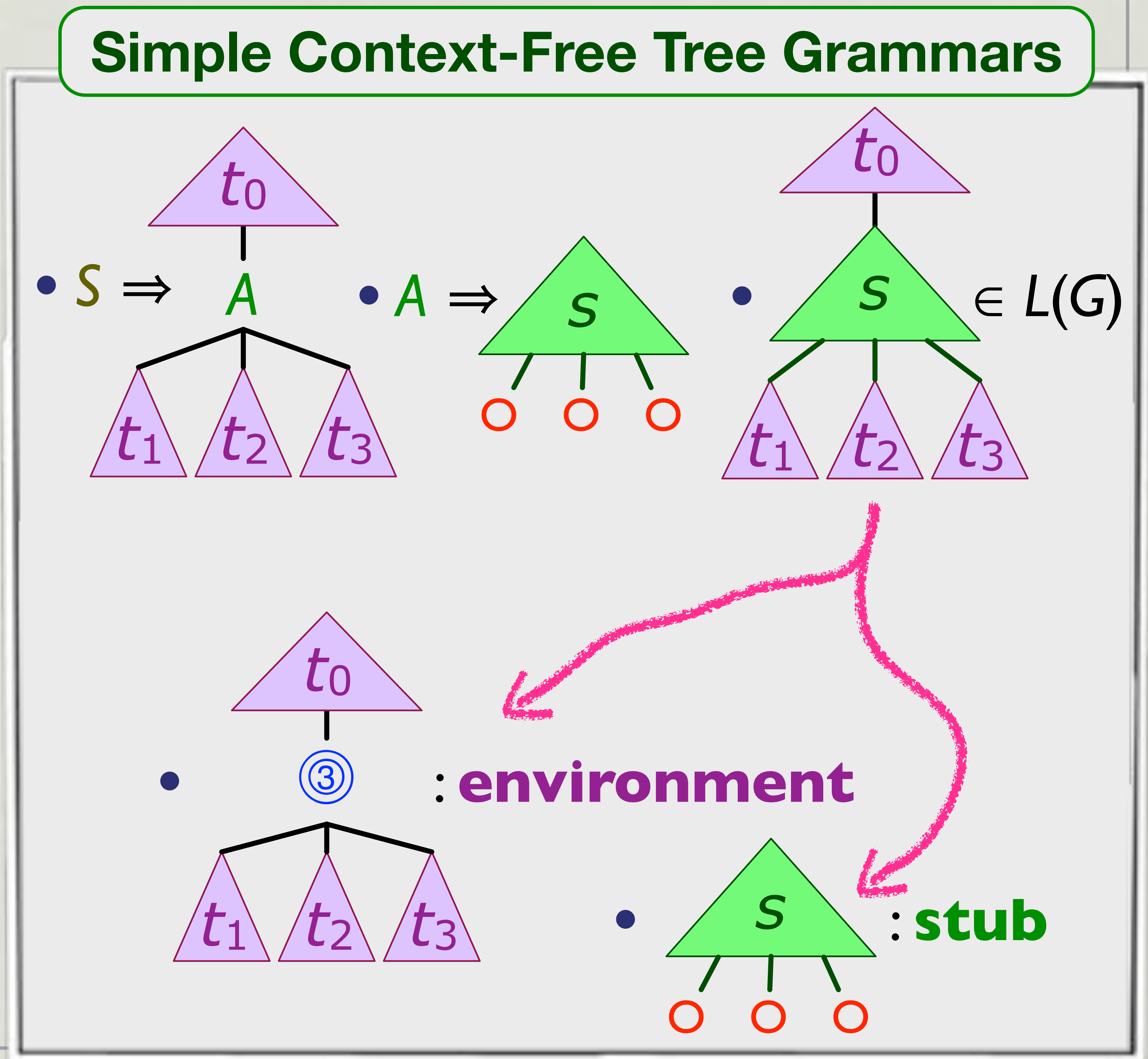
Substructure/Context Decomposition

CFGs

- $S \Rightarrow u_1 A u_2$
- $A \Rightarrow v$
- $u_1 v u_2 \in L(G)$

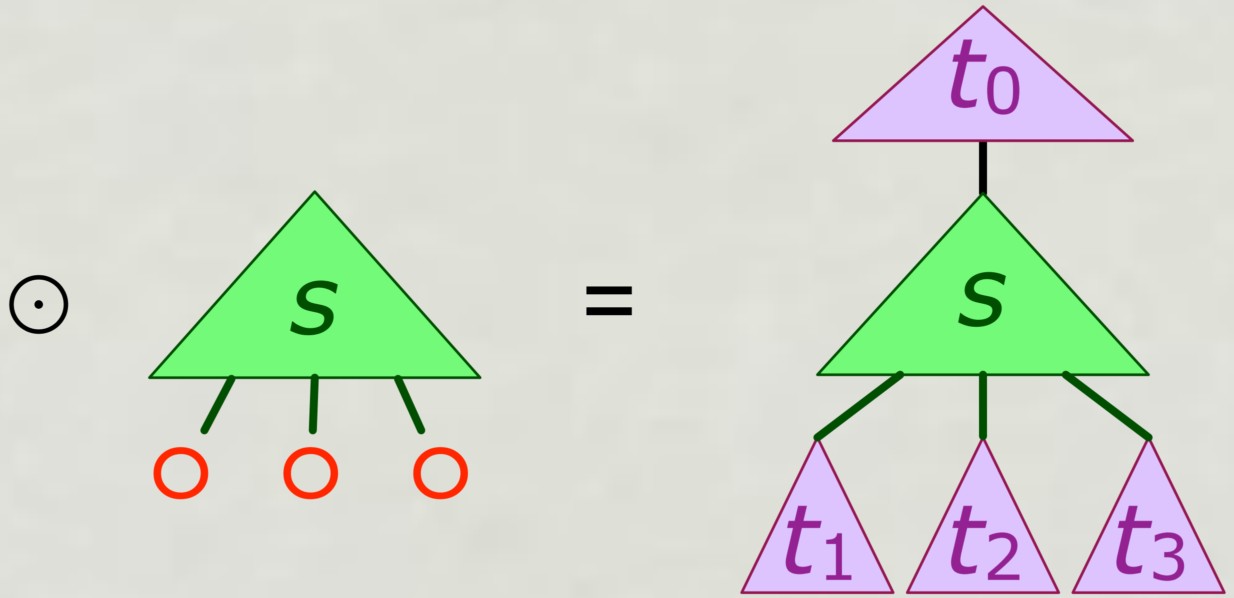
(u_1,u_2): **context**

v : **substring**



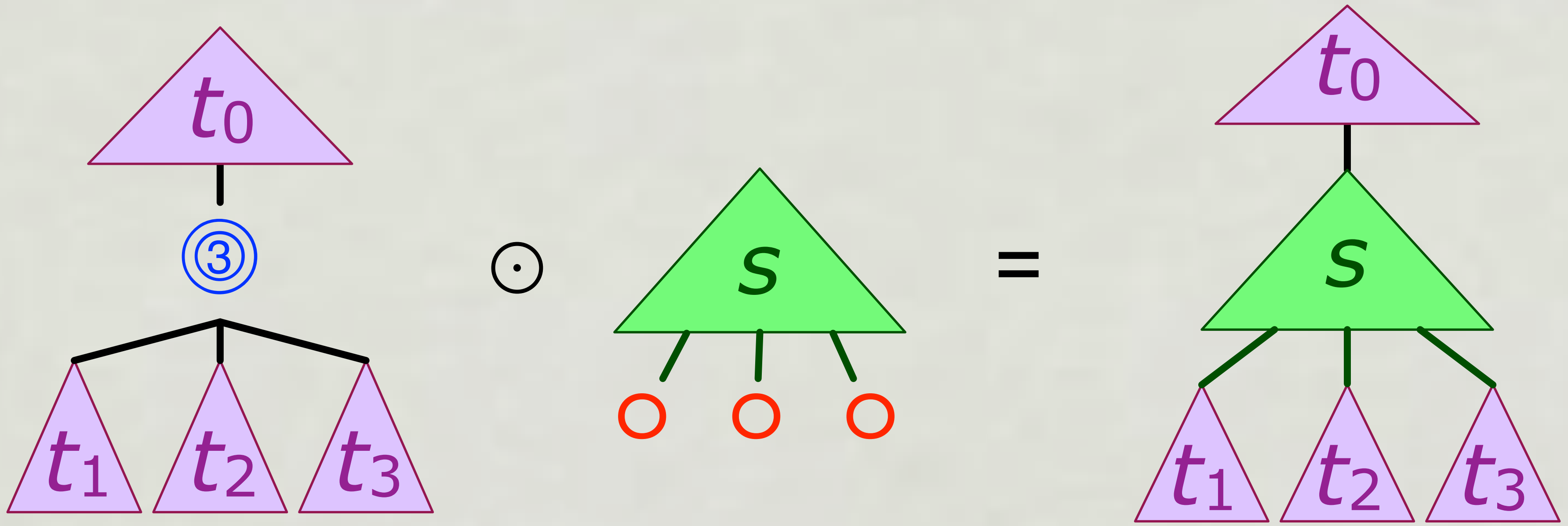
Composition/Decomposition

- *m-environment* e ... tree with a special symbol ⓜ of rank *m*
- *m-stub* s ... tree with *m* open leaves ○
- e ⊙ s .... substitute s for ⓜ in e



Composition/Decomposition

- *m-environment* e ... tree with a special symbol ⓜ of rank *m*
- *m-stub* s ... tree with *m* open leaves ○
- e $\odot$ s .... substitute s for ⓜ in e
- L_0/S = { e | e $\odot$ $S \subseteq L_0$ } : the set of tree-contexts for ***s***



Finite Kernel Property

- A set of stubs S_A is a ***p*-kernel** of L(G, A) (or of $A \in N$) iff
  iff L(G)/S_A = L(G)/L(A)
  (i.e., $E \odot S_A \in L(G) \;\Rightarrow\; E \odot \mathrm{L}(A) \subseteq L(G)$ for any environment E)

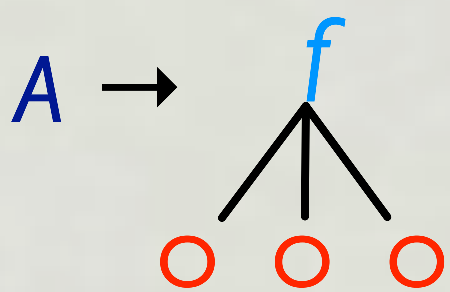
Finite Kernel Property

- A set of stubs S_A is a ***p*-kernel** of $L(G, A)$ (or of $A \in N$) iff
  iff $L(G)/S_A = L(G)/L(A)$
  (i.e., $E \odot S_A \in L(G) \Rightarrow E \odot L(A) \subseteq L(G)$ for any environment E)

- r-SCFTG G has the ***p*-Finite Kernel Property**
  iff every nonterminal admits a p-kernel

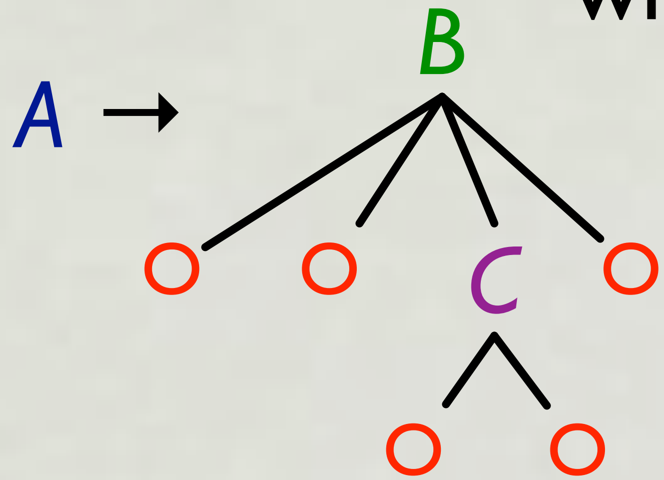
Chomsky Normal Form

- Every rule has one of the forms:
  - $A \to f\langle \circ,\ldots,\circ \rangle$ for $f \in \Sigma_k$ with $k = \mathrm{rnk}(f)$



  - $A \to B\langle \circ,\ldots,\circ,C\langle \circ,\ldots,\circ \rangle,\circ,\ldots,\circ \rangle$

    where $\mathrm{rank}(A) = \mathrm{rank}(B) + \mathrm{rank}(C) - 1$



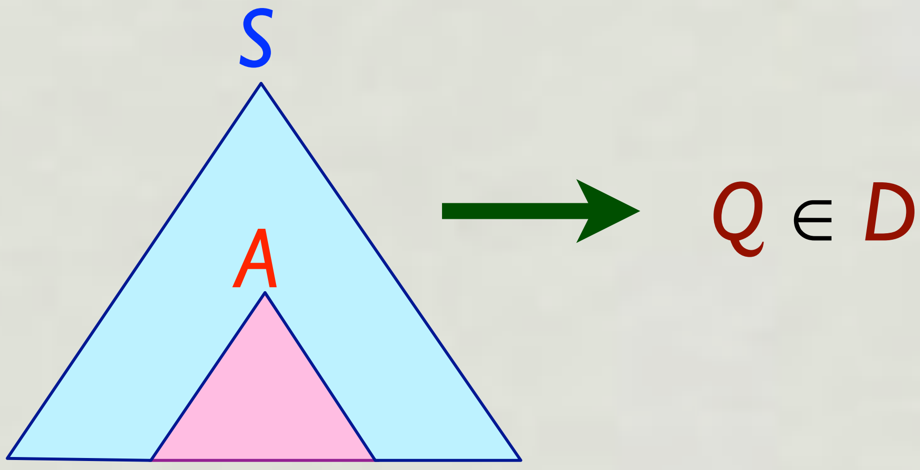
Learning r-SCFTGs with p-FKP

- $K_i \subseteq \mathrm{Stub}_i(D)$ for $i = 0,\dots,r$ for D positive data
- $X_i = \mathrm{Env}_i(D) = \{\, \boldsymbol{w} \mid \boldsymbol{w} \odot \boldsymbol{v} \in D$ for some $\boldsymbol{v} \in K_i \,\}$
- $G_{K,X}$: conjecture
  - $N_i = \{\, [\![S]\!] \mid S \subseteq K_i, |S| \leqq p \,\}$
  - ★ $[\![S]\!] \Rightarrow s$ for $L_0/S \subseteq L_0/s$
  - Initial Symbols: $\{\, [\![T]\!] \in N_0 \mid T \subseteq L_0 \,\}$
  - Rules
    - $[\![S_0]\!](\circ,\dots,\circ) \to [\![S_1]\!](\circ,\dots,\circ,[\![S_2]\!](\circ,\dots,\circ),\circ,\dots,\circ)$
      if $L_0/S_0[\circ,\dots,\circ] \cap X \subseteq L_0/S_1[\circ,\dots,\circ,S_2[\circ,\dots,\circ],\circ,\dots,\circ] \cap X$
    - $[\![S_0]\!](\circ,\dots,\circ) \to a(\circ,\dots,\circ)$ if $L_0/S_0 \cap X \subseteq L_0/a \cap X$

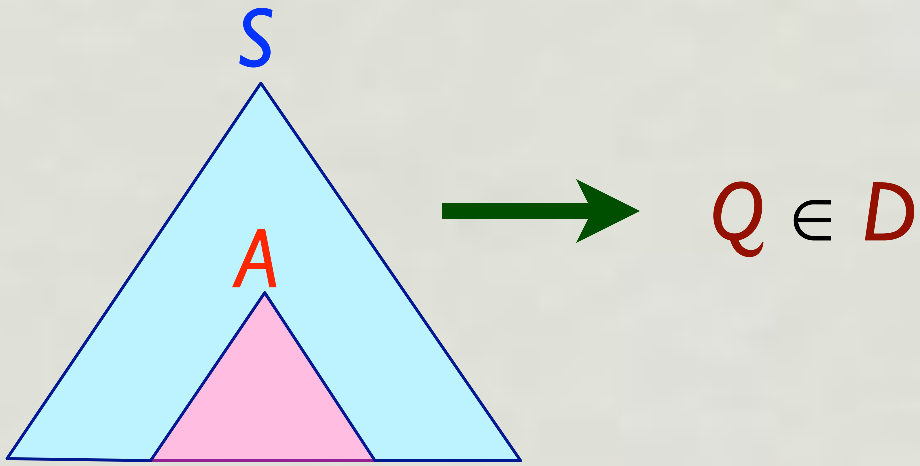
Context-Free Formalisms

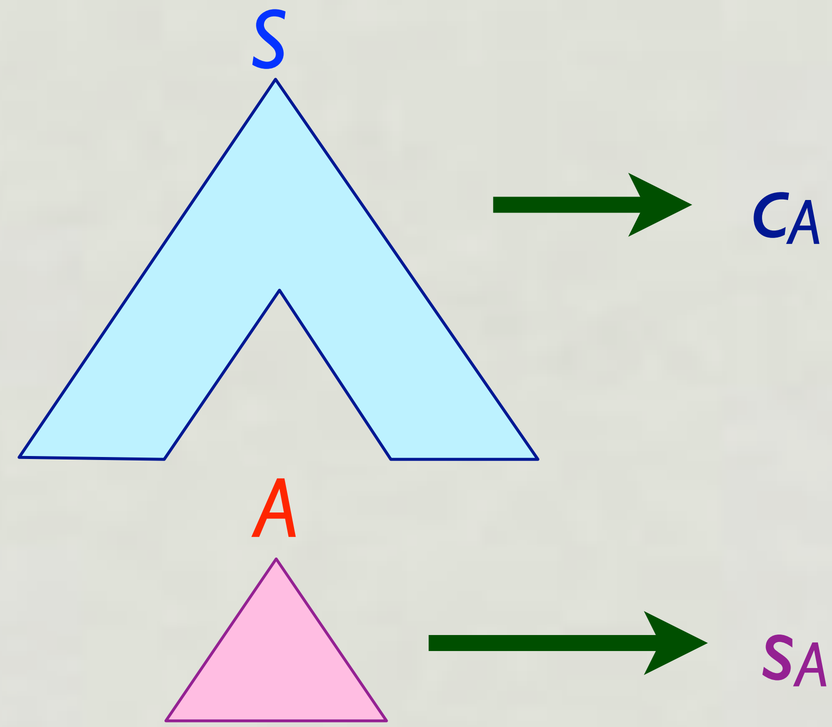
- Context-Free Grammars
- Multiple Context-Free Grammars
- (Multiple) Simple Context-Free Tree Grammars
- Simple Macro Grammars
- Hyper-Edge Replacement Grammars
- Linear Context-Free Lambda Grammars
  (2nd order ACG)
- etc.

- CF-derivation tree



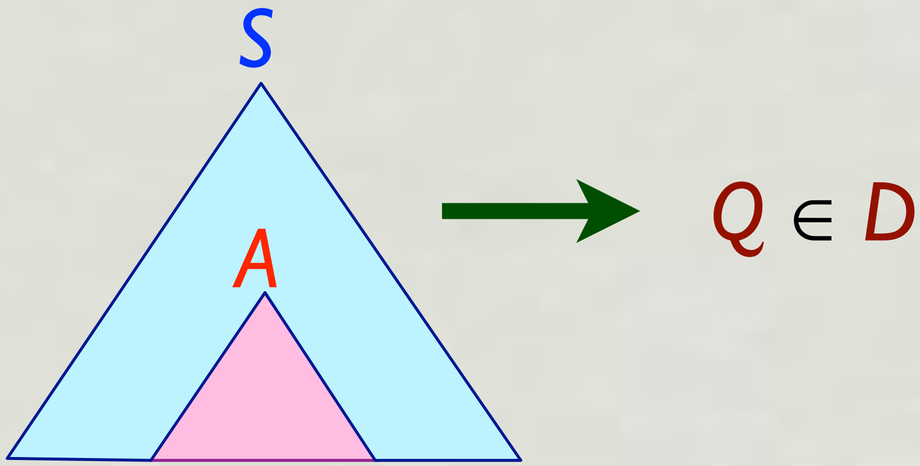
- CF-derivation tree

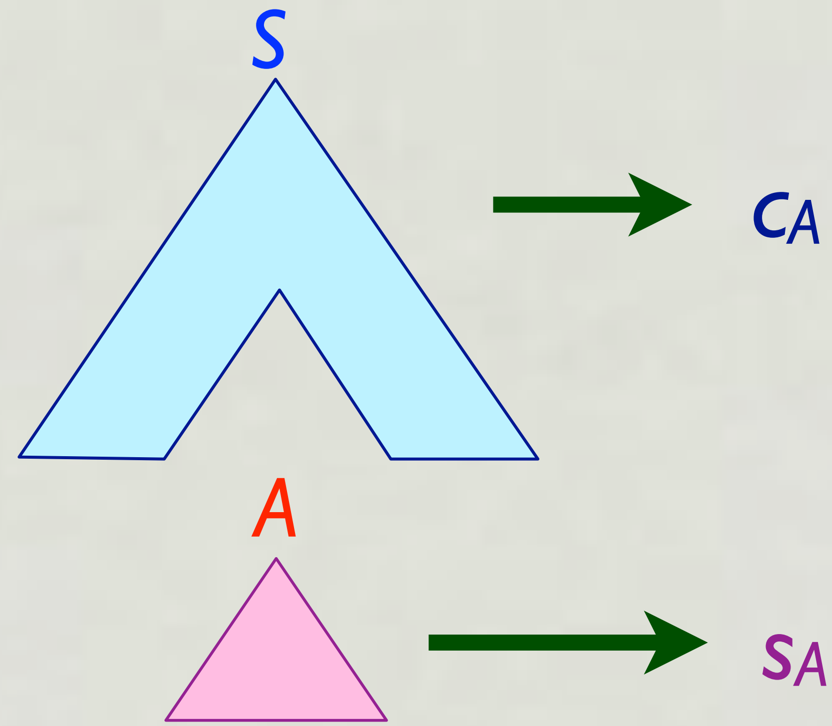




$$Q = c_A \odot s_A$$

- CF-derivation tree

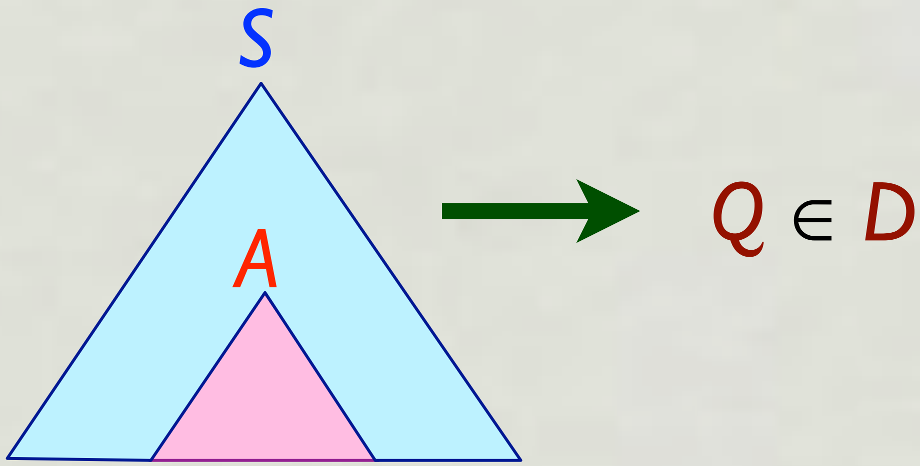


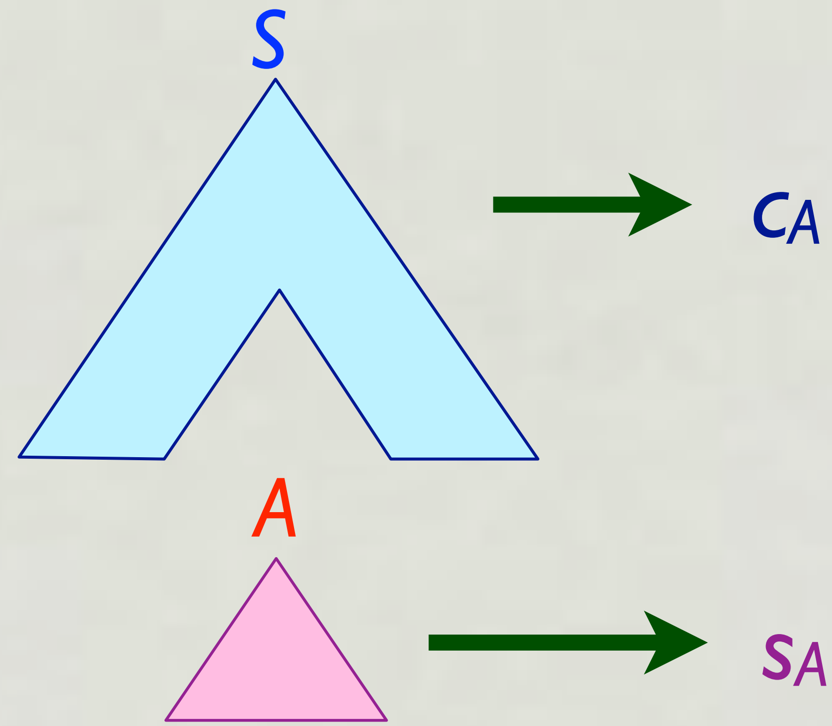


$$Q = c_A \odot s_A$$

- $\mathrm{Sub}_i(D) = \{\ s \mid c \odot_i s \in D \text{ for some } c\ \}$
- $\mathrm{Con}_i(D) = \{\ c \mid c \odot_i s \in D \text{ for some } s\ \}$

- CF-derivation tree





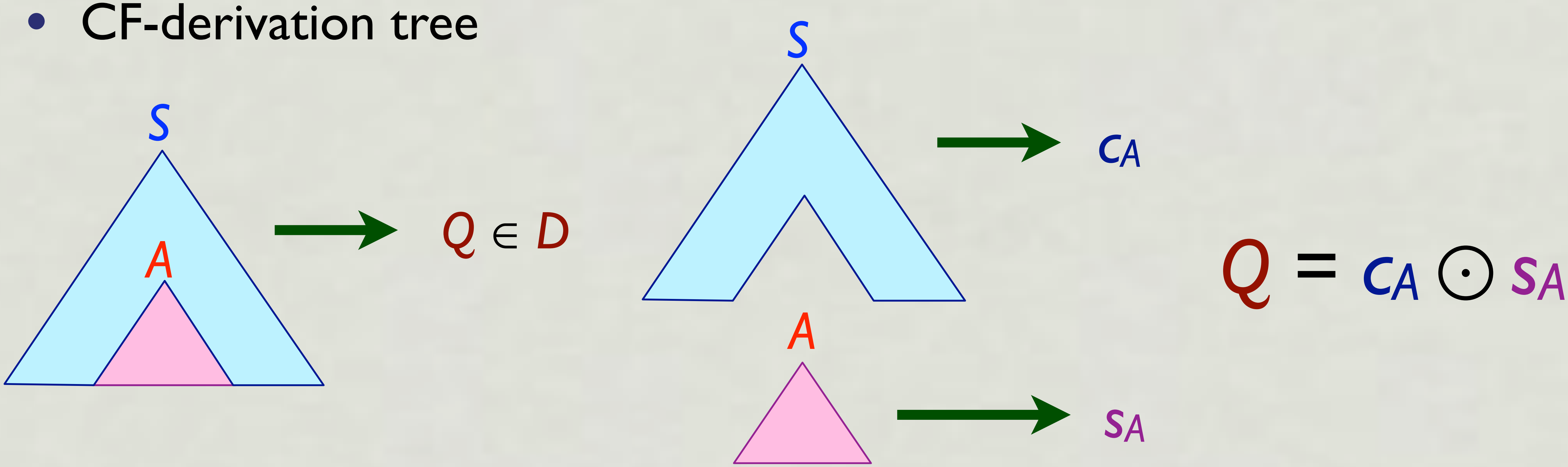
$$Q = c_A \odot s_A$$

- $\mathrm{Sub}_i(D) = \{\, s \mid c \odot_i s \in D$ for some $c \,\}$
- $\mathrm{Con}_i(D) = \{\, c \mid c \odot_i s \in D$ for some $s \,\}$

- CF-derivation rule

When $A \to \varphi(B,C)$ is used to derive Q, φ must be observable in Q

- CF-derivation tree



- $\mathrm{Sub}_i(D) = \{ s \mid c \odot_i s \in D \text{ for some } c \}$
- $\mathrm{Con}_i(D) = \{ c \mid c \odot_i s \in D \text{ for some } s \}$

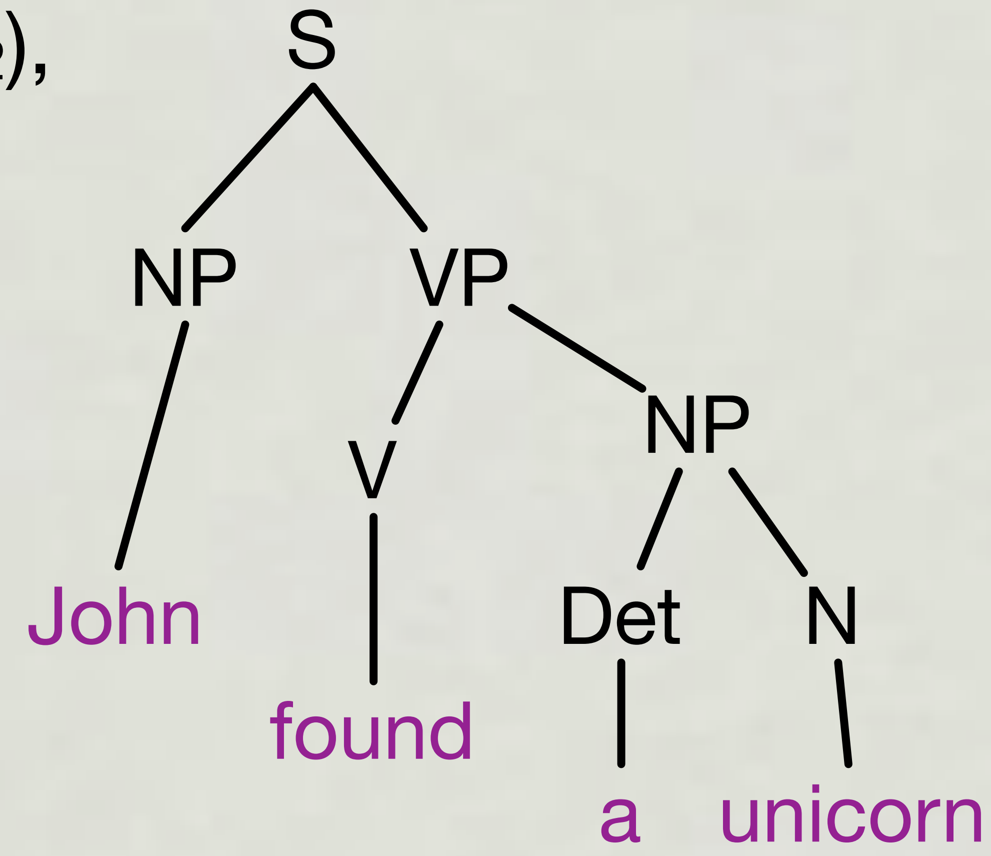
- CF-derivation rule

When $A \rightarrow \varphi(B,C)$ is used to derive Q, φ must be observable in Q

  - Construct all possible rules from those components
  - All correct rules should be obtained
  - All incorrect rules should be rejected

CFG with Montague Semantics

- $S(w_1w_2, Z_1Z_2)$:– $NP(w_1, Z_1)\ VP(w_2, Z_2)$,
  $VP(w_1w_2, \lambda x.Z_2(\lambda y.Z_1yx))$:– $V(w_1,Z_1)\ NP(w_2, Z_2)$,
  $NP(w_1w_2, Z_1Z_2)$:– $Det(w_1, X_1)\ N(w_2, Z_2)$,
  NP(John , $\lambda u.u$ **John**) :– ,
  V(found , $\lambda yz.$ **find** yz) :–,
  Det(a , $\lambda uv.$**Intersect** $u\ v$) :–,
  N(unicorn , $\lambda y.$ **unicorn** y) :–



- (John found a unicorn, **Intersect** ($\lambda y.$ **unicorn** y) ($\lambda y.$**find** y **John**))
- Semantics-driven learning

Copying : Non-linear λ-CFG ?

- Copying : Non-simple CFTG, Non-linear λ-CFG
  - Syntax: Yoruba
  - Semantics:
    - Det(a , $\lambda uv.$**Intersect** $u\ v$) :–
    - $\lambda uv.$**Intersect** $u\ v = \lambda uv.\exists(\lambda y.\wedge\ (uy)\ (vy))$
    - and, himself, etc.

Copying : Non-linear λ-CFG ?

- Copying : Non-simple CFTG, Non-linear λ-CFG
  - Syntax: Yoruba
  - Semantics:
    - Det(a , $\lambda uv.$**Intersect** $u\ v$) :–
    - $\lambda uv.$**Intersect** $u\ v = \lambda uv.\exists(\lambda y. \wedge (uy)\ (vy))$
    - and, himself, etc.
- Higher-order ACGs ?