Defining contexts in context-free grammars

Mikhail Barash   Alexander Okhotin

Department of Mathematics, University of Turku, Finland

6 March, A. D. 2012

Inductive descriptions of syntax

Inductive descriptions of syntax

- Mathematics, programming languages, etc.:
  - Any number is an *expression*.

Inductive descriptions of syntax

- Mathematics, programming languages, etc.:
  - Any number is an *expression*.
  - If e and e' are *expressions*, then so are $e + e'$, $e * e'$ and (e).

Inductive descriptions of syntax

- Mathematics, programming languages, etc.:
  - Any number is an *expression*.
  - If e and e' are *expressions*, then so are $e + e'$, $e * e'$ and (e).
- Natural languages:
  - A *subject* followed by a *predicate* is a *sentence*.
  - An *article* followed by an *adjective* and a *noun* is a *subject*.

Inductive descriptions of syntax

- Mathematics, programming languages, etc.:
  - Any number is an *expression*.
  - If e and e' are *expressions*, then so are $e + e'$, $e * e'$ and (e).
- Natural languages:
  - A *subject* followed by a *predicate* is a *sentence*.
  - An *article* followed by an *adjective* and a *noun* is a *subject*.
- Formal notation: $G = (\Sigma, N, R, S)$, where

Inductive descriptions of syntax

- Mathematics, programming languages, etc.:
  - Any number is an *expression*.
  - If e and e' are *expressions*, then so are $e + e'$, $e * e'$ and (e).
- Natural languages:
  - A *subject* followed by a *predicate* is a *sentence*.
  - An *article* followed by an *adjective* and a *noun* is a *subject*.
- Formal notation: $G = (\Sigma, N, R, S)$, where

  Σ: alphabet of the target language (a,b,c,...)

Inductive descriptions of syntax

- Mathematics, programming languages, etc.:
  - Any number is an *expression*.
  - If e and e' are *expressions*, then so are $e + e'$, $e * e'$ and (e).
- Natural languages:
  - A *subject* followed by a *predicate* is a *sentence*.
  - An *article* followed by an *adjective* and a *noun* is a *subject*.
- Formal notation: $G = (\Sigma, N, R, S)$, where

  Σ: alphabet of the target language ($\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots$)

  N: auxiliary symbols for syntactic properties, such as *expression* or *subject*.

Inductive descriptions of syntax

- Mathematics, programming languages, etc.:
  - Any number is an *expression*.
  - If e and e' are *expressions*, then so are $e + e'$, $e * e'$ and (e).
- Natural languages:
  - A *subject* followed by a *predicate* is a *sentence*.
  - An *article* followed by an *adjective* and a *noun* is a *subject*.
- Formal notation: $G = (\Sigma, N, R, S)$, where

  Σ: alphabet of the target language (a,b,c,...)

  N: auxiliary symbols for syntactic properties, such as *expression* or *subject*.

  - Each string $w \in \Sigma^*$ may have or not have each property $A \in N$.

Inductive descriptions of syntax

- Mathematics, programming languages, etc.:
  - Any number is an *expression*.
  - If e and e' are *expressions*, then so are $e + e'$, $e * e'$ and (e).
- Natural languages:
  - A *subject* followed by a *predicate* is a *sentence*.
  - An *article* followed by an *adjective* and a *noun* is a *subject*.
- Formal notation: $G = (\Sigma, N, R, S)$, where

  Σ: alphabet of the target language (a,b,c,. . .)

  N: auxiliary symbols for syntactic properties, such as *expression* or *subject*.

  - Each string $w \in \Sigma^*$ may have or not have each property $A \in N$.

  R: set of rules, each of the form

$$A \to X_1 \ldots X_\ell \quad (A \in N,\ \ell \geqslant 0,\ X_i \in \Sigma \cup N) \qquad (*)$$

Inductive descriptions of syntax

- Mathematics, programming languages, etc.:
  - Any number is an *expression*.
  - If e and e' are *expressions*, then so are $e + e'$, $e * e'$ and (e).
- Natural languages:
  - A *subject* followed by a *predicate* is a *sentence*.
  - An *article* followed by an *adjective* and a *noun* is a *subject*.
- Formal notation: $G = (\Sigma, N, R, S)$, where

  Σ: alphabet of the target language (a,b,c,...)

  N: auxiliary symbols for syntactic properties, such as *expression* or *subject*.
  - ★ Each string $w \in \Sigma^*$ may have or not have each property $A \in N$.

  R: set of rules, each of the form

$$A \to X_1 \dots X_\ell \quad (A \in N,\ \ell \geqslant 0,\ X_i \in \Sigma \cup N) \qquad (*)$$

$S \in N$: the set of well-formed **s**entences.

Inductive descriptions of syntax

- Mathematics, programming languages, etc.:
  - Any number is an *expression*.
  - If e and e' are *expressions*, then so are $e + e'$, $e * e'$ and (e).
- Natural languages:
  - A *subject* followed by a *predicate* is a *sentence*.
  - An *article* followed by an *adjective* and a *noun* is a *subject*.
- Formal notation: $G = (\Sigma, N, R, S)$, where

  Σ: alphabet of the target language (a,b,c,...)

  N: auxiliary symbols for syntactic properties, such as *expression* or *subject*.

  - ★ Each string $w \in \Sigma^*$ may have or not have each property $A \in N$.

  R: set of rules, each of the form

$$A \to X_1 \dots X_\ell \quad (A \in N,\ \ell \geqslant 0,\ X_i \in \Sigma \cup N) \qquad (*)$$

$S \in N$: the set of well-formed **s**entences.

(*) If $w \in \Sigma^*$ is representable as $X_1 \cdot \ldots \cdot X_\ell$, then w has the property A.

Chomsky's formalization of grammars

Chomsky's formalization of grammars

- Rewriting strings over $(\Sigma \cup N)^*$, grammar rules as "productions":

$$\mu A \nu \Longrightarrow \mu \alpha \nu \quad (A \to \alpha \in R)$$

Chomsky's formalization of grammars

- Rewriting strings over $(\Sigma \cup N)^*$, grammar rules as "productions":

$$\mu A \nu \Longrightarrow \mu \alpha \nu \quad (A \to \alpha \in R)$$

- $w \in \Sigma^*$ has the property $A \in N$, if $A \Longrightarrow \ldots \Longrightarrow w$.

Chomsky's formalization of grammars

- Rewriting strings over $(\Sigma \cup N)^*$, grammar rules as "productions":

$$\mu A \nu \Longrightarrow \mu \alpha \nu \quad (A \to \alpha \in R)$$

- $w \in \Sigma^*$ has the property $A \in N$, if $A \Longrightarrow \ldots \Longrightarrow w$.
- $a \in \Sigma$: "terminal symbols", $A \in N$: "nonterminal symbols".

Chomsky's formalization of grammars

- Rewriting strings over $(\Sigma \cup N)^*$, grammar rules as "productions":

$$\mu A \nu \Longrightarrow \mu \alpha \nu \quad (A \to \alpha \in R)$$

- $w \in \Sigma^*$ has the property $A \in N$, if $A \Longrightarrow \ldots \Longrightarrow w$.
- $a \in \Sigma$: "terminal symbols", $A \in N$: "nonterminal symbols".

✓ Correctly defines inductive phrase structure.

Chomsky's formalization of grammars

- Rewriting strings over $(\Sigma \cup N)^*$, grammar rules as "productions":

$$\mu A \nu \Longrightarrow \mu \alpha \nu \quad (A \to \alpha \in R)$$

- $w \in \Sigma^*$ has the property $A \in N$, if $A \Longrightarrow \ldots \Longrightarrow w$.
- $a \in \Sigma$: "terminal symbols", $A \in N$: "nonterminal symbols".

✓ Correctly defines inductive phrase structure.

Second model: "context-sensitive grammars":

Chomsky's formalization of grammars

- Rewriting strings over $(\Sigma \cup N)^*$, grammar rules as "productions":

$$\mu A \nu \Longrightarrow \mu \alpha \nu \quad (A \to \alpha \in R)$$

- $w \in \Sigma^*$ has the property $A \in N$, if $A \Longrightarrow \ldots \Longrightarrow w$.
- $a \in \Sigma$: "terminal symbols", $A \in N$: "nonterminal symbols".

✓ Correctly defines inductive phrase structure.

Second model: "context-sensitive grammars":

- Chomsky's intention: **rule applicable in a context**.

Chomsky's formalization of grammars

- Rewriting strings over $(\Sigma \cup N)^*$, grammar rules as "productions":

$$\mu A \nu \Longrightarrow \mu \alpha \nu \quad (A \to \alpha \in R)$$

- $w \in \Sigma^*$ has the property $A \in N$, if $A \Longrightarrow \ldots \Longrightarrow w$.
- $a \in \Sigma$: "terminal symbols", $A \in N$: "nonterminal symbols".

✓ Correctly defines inductive phrase structure.

Second model: "context-sensitive grammars":

- Chomsky's intention: **rule applicable in a context**.
- Chomsky's formalization: production $\beta A \gamma \to \beta \alpha \gamma$, rewriting $\mu \beta A \gamma \nu \Longrightarrow \mu \beta \alpha \gamma \nu$.

Chomsky's formalization of grammars

- Rewriting strings over $(\Sigma \cup N)^*$, grammar rules as "productions":

$$\mu A \nu \Longrightarrow \mu \alpha \nu \quad (A \to \alpha \in R)$$

- $w \in \Sigma^*$ has the property $A \in N$, if $A \Longrightarrow \ldots \Longrightarrow w$.
- $a \in \Sigma$: "terminal symbols", $A \in N$: "nonterminal symbols".

✓ Correctly defines inductive phrase structure.

Second model: "context-sensitive grammars":

- Chomsky's intention: **rule applicable in a context**.
- Chomsky's formalization: production $\beta A \gamma \to \beta \alpha \gamma$, rewriting $\mu \beta A \gamma \nu \Longrightarrow \mu \beta \alpha \gamma \nu$.
- Equivalent to NSPACE(n).

Chomsky's formalization of grammars

- Rewriting strings over $(\Sigma \cup N)^*$, grammar rules as "productions":

$$\mu A \nu \Longrightarrow \mu \alpha \nu \quad (A \to \alpha \in R)$$

- $w \in \Sigma^*$ has the property $A \in N$, if $A \Longrightarrow \ldots \Longrightarrow w$.
- $a \in \Sigma$: "terminal symbols", $A \in N$: "nonterminal symbols".

✓ Correctly defines inductive phrase structure.

Second model: "context-sensitive grammars":

- Chomsky's intention: **rule applicable in a context**.
- Chomsky's formalization: production $\beta A \gamma \to \beta \alpha \gamma$, rewriting $\mu \beta A \gamma \nu \Longrightarrow \mu \beta \alpha \gamma \nu$.
- Equivalent to NSPACE(n).
  - $A \in N$ are no longer syntactic properties, but bits in the memory!

Chomsky's formalization of grammars

- Rewriting strings over $(\Sigma \cup N)^*$, grammar rules as "productions":

$$\mu A \nu \Longrightarrow \mu \alpha \nu \quad (A \to \alpha \in R)$$

- $w \in \Sigma^*$ has the property $A \in N$, if $A \Longrightarrow \ldots \Longrightarrow w$.
- $a \in \Sigma$: "terminal symbols", $A \in N$: "nonterminal symbols".

✓ Correctly defines inductive phrase structure.

Second model: "context-sensitive grammars":

- Chomsky's intention: **rule applicable in a context**.
- Chomsky's formalization: production $\beta A \gamma \to \beta \alpha \gamma$, rewriting $\mu \beta A \gamma \nu \Longrightarrow \mu \beta \alpha \gamma \nu$.
- Equivalent to NSPACE(n).
  - $A \in N$ are no longer syntactic properties, but bits in the memory!

✓ Wrong formalization.

Chomsky's formalization of grammars

- Rewriting strings over $(\Sigma \cup N)^*$, grammar rules as "productions":

$$\mu A \nu \Longrightarrow \mu \alpha \nu \quad (A \to \alpha \in R)$$

- $w \in \Sigma^*$ has the property $A \in N$, if $A \Longrightarrow \ldots \Longrightarrow w$.
- $a \in \Sigma$: "terminal symbols", $A \in N$: "nonterminal symbols".

✓ Correctly defines inductive phrase structure.

Second model: "context-sensitive grammars":

- Chomsky's intention: **rule applicable in a context**.
- Chomsky's formalization: production $\beta A \gamma \to \beta \alpha \gamma$, rewriting $\mu \beta A \gamma \nu \Longrightarrow \mu \beta \alpha \gamma \nu$.
- Equivalent to NSPACE(n).
  - $A \in N$ are no longer syntactic properties, but bits in the memory!

✓ Wrong formalization.

The idea deserves a correct implementation!

Conjunctive grammars (Okhotin, 2000)

Conjunctive grammars (Okhotin, 2000)

$G = (\Sigma, N, P, S)$, rules of the form

$$A \to \alpha_1 \& \ldots \& \alpha_m \quad (\alpha_1, \ldots, \alpha_m \in (\Sigma \cup N)^*)$$

"If w is representable as each α_i, then w has the property A"

Conjunctive grammars (Okhotin, 2000)

$G = (\Sigma, N, P, S)$, rules of the form

$$A \to \alpha_1 \& \ldots \& \alpha_m \quad (\alpha_1, \ldots, \alpha_m \in (\Sigma \cup N)^*)$$

"If w is representable as each α_i, then w has the property A"

- Two equivalent formalizations: term rewriting, language equations.

Conjunctive grammars (Okhotin, 2000)

$G = (\Sigma, N, P, S)$, rules of the form

$$A \to \alpha_1 \& \ldots \& \alpha_m \quad (\alpha_1, \ldots, \alpha_m \in (\Sigma \cup N)^*)$$

"If w is representable as each α_i, then w has the property A"

- Two equivalent formalizations: term rewriting, language equations.
- Essentially "context-free" descriptions: no contexts involved.

Conjunctive grammars (Okhotin, 2000)

$G = (\Sigma, N, P, S)$, rules of the form

$$A \to \alpha_1 \& \ldots \& \alpha_m \quad (\alpha_1, \ldots, \alpha_m \in (\Sigma \cup N)^*)$$

"If w is representable as each α_i, then w has the property A"

- Two equivalent formalizations: term rewriting, language equations.
- Essentially "context-free" descriptions: no contexts involved.
- Can generate quite a lot.

Conjunctive grammars (Okhotin, 2000)

$G = (\Sigma, N, P, S)$, rules of the form

$$A \to \alpha_1 \& \ldots \& \alpha_m \quad (\alpha_1, \ldots, \alpha_m \in (\Sigma \cup N)^*)$$

"If w is representable as each α_i, then w has the property A"

- Two equivalent formalizations: term rewriting, language equations.
- Essentially "context-free" descriptions: no contexts involved.
- Can generate quite a lot.
- Parsing in time $O(n^\omega)$, with $\omega < 3$.

Conjunctive grammars (Okhotin, 2000)

$G = (\Sigma, N, P, S)$, rules of the form

$$A \to \alpha_1 \& \ldots \& \alpha_m \quad (\alpha_1, \ldots, \alpha_m \in (\Sigma \cup N)^*)$$

"If w is representable as each α_i, then w has the property A"

- Two equivalent formalizations: term rewriting, language equations.
- Essentially "context-free" descriptions: no contexts involved.
- Can generate quite a lot.
- Parsing in time $O(n^\omega)$, with $\omega < 3$.
- ... and other properties.

Conjunctive grammars (Okhotin, 2000)

$G = (\Sigma, N, P, S)$, rules of the form

$$A \to \alpha_1 \& \ldots \& \alpha_m \quad (\alpha_1, \ldots, \alpha_m \in (\Sigma \cup N)^*)$$

"If w is representable as each α_i, then w has the property A"

- Two equivalent formalizations: term rewriting, language equations.
- Essentially "context-free" descriptions: no contexts involved.
- Can generate quite a lot.
- Parsing in time $O(n^\omega)$, with $\omega < 3$.
- ...and other properties.

Successful extension of inductive definitions beyond context-free grammars.

Grammars with one-sided contexts

$G = (\Sigma, N, R, S)$, with rules of the form

$$A \to \alpha_1 \,\&\, \dots \,\&\, \alpha_k \,\&\, \lhd\beta_1 \,\&\, \dots \,\&\, \lhd\beta_m \,\&\, \trianglelefteq\gamma_1 \,\&\, \dots \,\&\, \trianglelefteq\gamma_n$$

Grammars with one-sided contexts

$G = (\Sigma, N, R, S)$, with rules of the form

$$A \to \alpha_1 \,\&\, \dots \,\&\, \alpha_k \,\&\, \lhd\beta_1 \,\&\, \dots \,\&\, \lhd\beta_m \,\&\, \unlhd\gamma_1 \,\&\, \dots \,\&\, \unlhd\gamma_n$$

Assume that a string $v \in \Sigma^*$ with left context $u \in \Sigma^*$ satisfies

- v is of the form α_i, for each i;

Grammars with one-sided contexts

$G = (\Sigma, N, R, S)$, with rules of the form

$$A \to \alpha_1 \,\&\, \dots \,\&\, \alpha_k \,\&\, \lhd\beta_1 \,\&\, \dots \,\&\, \lhd\beta_m \,\&\, \trianglelefteq\gamma_1 \,\&\, \dots \,\&\, \trianglelefteq\gamma_n$$

Assume that a string $v \in \Sigma^*$ with left context $u \in \Sigma^*$ satisfies

- v is of the form α_i, for each i;
- u is of the form β_i, for each i;

Grammars with one-sided contexts

$G = (\Sigma, N, R, S)$, with rules of the form

$$A \to \alpha_1 \,\&\, \ldots \,\&\, \alpha_k \,\&\, \lhd\beta_1 \,\&\, \ldots \,\&\, \lhd\beta_m \,\&\, \trianglelefteq\gamma_1 \,\&\, \ldots \,\&\, \trianglelefteq\gamma_n$$

Assume that a string $v \in \Sigma^*$ with left context $u \in \Sigma^*$ satisfies

- v is of the form α_i, for each i;
- u is of the form β_i, for each i;
- uv is of the form γ_i, for each i.

Grammars with one-sided contexts

$G = (\Sigma, N, R, S)$, with rules of the form

$$A \to \alpha_1 \,\&\, \dots \,\&\, \alpha_k \,\&\, \lhd\beta_1 \,\&\, \dots \,\&\, \lhd\beta_m \,\&\, \trianglelefteq\gamma_1 \,\&\, \dots \,\&\, \trianglelefteq\gamma_n$$

Assume that a string $v \in \Sigma^*$ with left context $u \in \Sigma^*$ satisfies

- v is of the form α_i, for each i;
- u is of the form β_i, for each i;
- uv is of the form γ_i, for each i.

Then v in context u has the property A.

Grammars with one-sided contexts

$G = (\Sigma, N, R, S)$, with rules of the form

$$A \to \alpha_1 \,\&\, \dots \,\&\, \alpha_k \,\&\, \lhd\beta_1 \,\&\, \dots \,\&\, \lhd\beta_m \,\&\, \unlhd\gamma_1 \,\&\, \dots \,\&\, \unlhd\gamma_n$$

Assume that a string $v \in \Sigma^*$ with left context $u \in \Sigma^*$ satisfies

- v is of the form α_i, for each i;
- u is of the form β_i, for each i;
- uv is of the form γ_i, for each i.

Then v in context u has the property A.

Grammar for $\{ a^n b^n c^n d^n \mid n \geqslant 0 \}$

$$\begin{aligned} S &\to aSd \mid bSc \mid \varepsilon \,\&\, \lhd A \\ A &\to aAb \mid \varepsilon \end{aligned}$$

Grammars with one-sided contexts

$G = (\Sigma, N, R, S)$, with rules of the form

$$A \to \alpha_1 \,\&\, \ldots \,\&\, \alpha_k \,\&\, \lhd\beta_1 \,\&\, \ldots \,\&\, \lhd\beta_m \,\&\, \trianglelefteq\gamma_1 \,\&\, \ldots \,\&\, \trianglelefteq\gamma_n$$

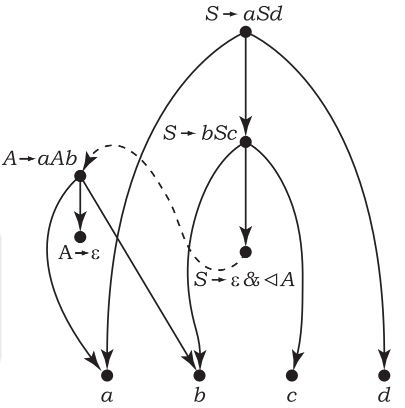
Assume that a string $v \in \Sigma^*$ with left context $u \in \Sigma^*$ satisfies

- v is of the form α_i, for each i;
- u is of the form β_i, for each i;
- uv is of the form γ_i, for each i.

Then v in context u has the property A.

Grammar for $\{\, a^n b^n c^n d^n \mid n \geqslant 0 \,\}$

$$S \to aSd \mid bSc \mid \varepsilon \,\&\, \lhd A$$
$$A \to aAb \mid \varepsilon$$



Grammars with one-sided contexts

$G = (\Sigma, N, R, S)$, with rules of the form

$$A \to \alpha_1 \,\&\, \ldots \,\&\, \alpha_k \,\&\, \lhd\beta_1 \,\&\, \ldots \,\&\, \lhd\beta_m \,\&\, \trianglelefteq\gamma_1 \,\&\, \ldots \,\&\, \trianglelefteq\gamma_n$$

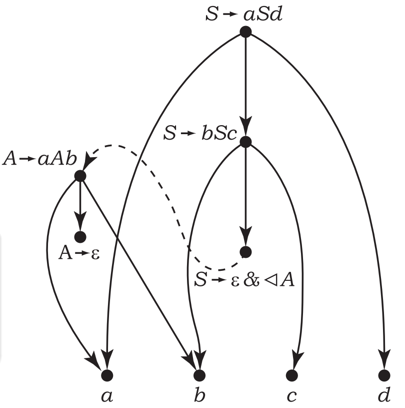
Assume that a string $v \in \Sigma^*$ with left context $u \in \Sigma^*$ satisfies

- v is of the form α_i, for each i;
- u is of the form β_i, for each i;
- uv is of the form γ_i, for each i.

Then v in context u has the property A.

Grammar for $\{ a^n b^n c^n d^n \mid n \geqslant 0 \}$

$$S \to aSd \mid bSc \mid \varepsilon \,\&\, \lhd A$$
$$A \to aAb \mid \varepsilon$$



✓ More involved example: "declaration before or after use".

Definition by deduction

Elementary proposition $[\alpha, \langle u \rangle v]$.

"a string $v \in \Sigma^*$ in left context $u \in \Sigma^*$ has the property $\alpha \in (\Sigma \cup N)^*$".

Definition by deduction

Elementary proposition $[\alpha, \langle u \rangle v]$.

"a string $v \in \Sigma^*$ in left context $u \in \Sigma^*$ has the property $\alpha \in (\Sigma \cup N)^*$".

Axiom scheme:

$$\vdash_G [a, \langle x \rangle a] \quad \text{(for all } a \in \Sigma \text{ and } x \in \Sigma^*\text{)}.$$

Definition by deduction

Elementary proposition $[\alpha, \langle u \rangle v]$.

"a string $v \in \Sigma^*$ in left context $u \in \Sigma^*$ has the property $\alpha \in (\Sigma \cup N)^*$".

Axiom scheme:

$$\vdash_G [a, \langle x \rangle a] \quad \text{(for all } a \in \Sigma \text{ and } x \in \Sigma^*\text{)}.$$

Each rule in R is a scheme for deduction rules.

Definition by deduction

Elementary proposition $[\alpha, \langle u \rangle v]$.

"a string $v \in \Sigma^*$ in left context $u \in \Sigma^*$ has the property $\alpha \in (\Sigma \cup N)^*$".

Axiom scheme:

$$\vdash_G [a, \langle x \rangle a] \quad (\text{for all } a \in \Sigma \text{ and } x \in \Sigma^*).$$

Each rule in R is a scheme for deduction rules. E.g., $A \to BC \& \lhd D$:

$$[B, \langle x \rangle u], [C, \langle xu \rangle v], [D, \langle \varepsilon \rangle x] \vdash_G [A, \langle x \rangle uv] \qquad (\text{for } x, u, v \in \Sigma^*).$$

Definition by deduction

Elementary proposition $[\alpha, \langle u \rangle v]$.

"a string $v \in \Sigma^*$ in left context $u \in \Sigma^*$ has the property $\alpha \in (\Sigma \cup N)^*$".

Axiom scheme:

$$\vdash_G [a, \langle x \rangle a] \quad (\text{for all } a \in \Sigma \text{ and } x \in \Sigma^*).$$

Each rule in R is a scheme for deduction rules. E.g., $A \to BC \& \lhd D$:

$$[B, \langle x \rangle u], [C, \langle xu \rangle v], [D, \langle \varepsilon \rangle x] \vdash_G [A, \langle x \rangle uv] \qquad (\text{for } x, u, v \in \Sigma^*).$$

Define

$$L_G(A) = \{ \langle u \rangle v \mid \vdash_G [A, \langle u \rangle v] \},$$
$$L(G) = \{ w \mid \vdash_G [S, \langle \varepsilon \rangle w] \}.$$

Definition by deduction

Elementary proposition $[\alpha, \langle u \rangle v]$.

"a string $v \in \Sigma^*$ in left context $u \in \Sigma^*$ has the property $\alpha \in (\Sigma \cup N)^*$".

Axiom scheme:

$$\vdash_G [a, \langle x \rangle a] \quad (\text{for all } a \in \Sigma \text{ and } x \in \Sigma^*).$$

Each rule in R is a scheme for deduction rules. E.g., $A \to BC \& \lhd D$:

$$[B, \langle x \rangle u], [C, \langle xu \rangle v], [D, \langle \varepsilon \rangle x] \vdash_G [A, \langle x \rangle uv] \qquad (\text{for } x, u, v \in \Sigma^*).$$

Define

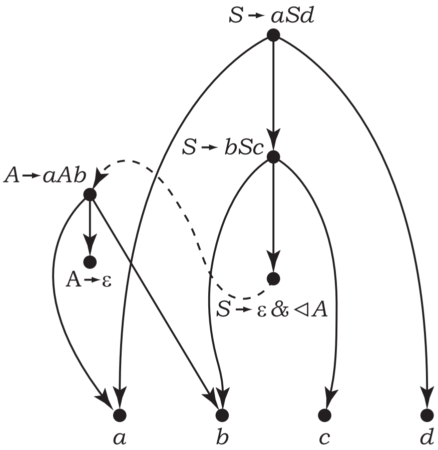
$$L_G(A) = \{ \langle u \rangle v \mid \vdash_G [A, \langle u \rangle v] \},$$
$$L(G) = \{ w \mid \vdash_G [S, \langle \varepsilon \rangle w] \}.$$

- A proof tree is a parse tree.

Example revisited

Grammar for $\{ a^n b^n c^n d^n \mid n \geqslant 0 \}$

$$S \rightarrow aSd \mid bSc \mid \varepsilon \,\&\, \lhd A$$
$$A \rightarrow aAb \mid \varepsilon$$

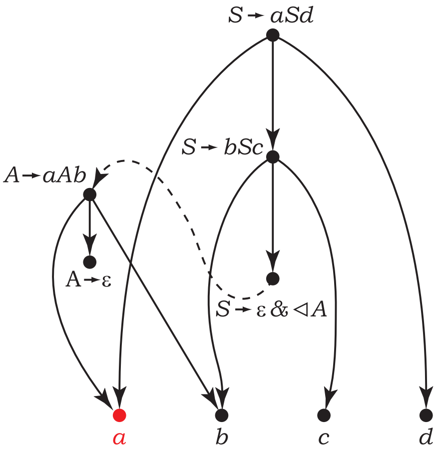


Example revisited

Grammar for $\{ a^n b^n c^n d^n \mid n \geqslant 0 \}$

$$S \rightarrow aSd \mid bSc \mid \varepsilon \,\&\, \lhd A$$
$$A \rightarrow aAb \mid \varepsilon$$

$\vdash [a, \langle \varepsilon \rangle a]$ *(axiom)*



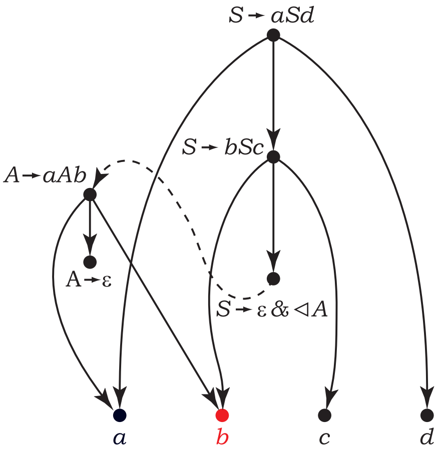
Example revisited

Grammar for $\{ a^n b^n c^n d^n \mid n \geqslant 0 \}$

$$S \rightarrow aSd \mid bSc \mid \varepsilon \,\&\, \lhd A$$
$$A \rightarrow aAb \mid \varepsilon$$

$\vdash [a, \langle \varepsilon \rangle a]$ (*axiom*)

$\vdash [b, \langle a \rangle b]$ (*axiom*)



Example revisited

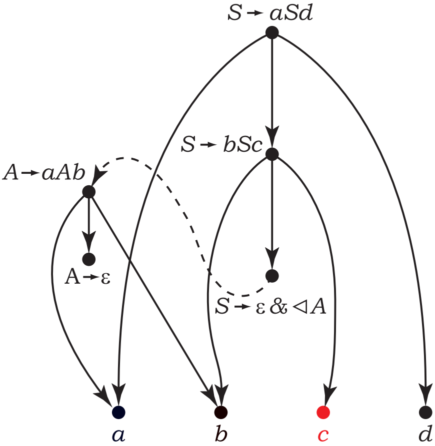
Grammar for $\{ a^n b^n c^n d^n \mid n \geqslant 0 \}$

$$S \rightarrow aSd \mid bSc \mid \varepsilon \,\&\, \lhd A$$
$$A \rightarrow aAb \mid \varepsilon$$

$\vdash [a, \langle\varepsilon\rangle a]$ (*axiom*)

$\vdash [b, \langle a\rangle b]$ (*axiom*)

$\vdash [c, \langle ab\rangle c]$ (*axiom*)



Example revisited

Grammar for $\{ a^n b^n c^n d^n \mid n \geqslant 0 \}$

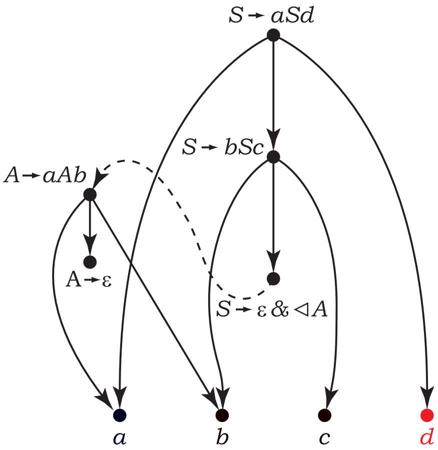
$$S \rightarrow aSd \mid bSc \mid \varepsilon \,\&\, \lhd A$$
$$A \rightarrow aAb \mid \varepsilon$$

$\vdash [a, \langle\varepsilon\rangle a]$ (*axiom*)

$\vdash [b, \langle a\rangle b]$ (*axiom*)

$\vdash [c, \langle ab\rangle c]$ (*axiom*)

$\vdash [d, \langle abc\rangle d]$ (*axiom*)



Example revisited

Grammar for $\{ a^n b^n c^n d^n \mid n \geqslant 0 \}$

$$S \rightarrow aSd \mid bSc \mid \varepsilon \,\&\, \lhd A$$
$$A \rightarrow aAb \mid \varepsilon$$

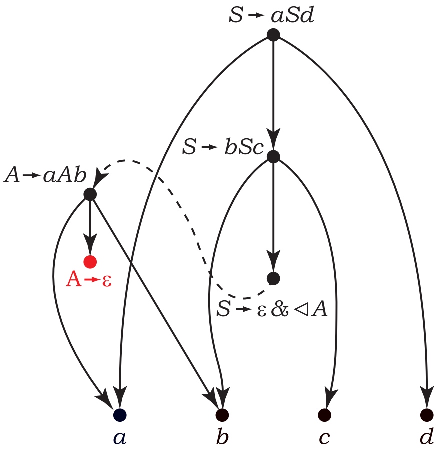
$\vdash [a, \langle \varepsilon \rangle a]$ $(axiom)$

$\vdash [b, \langle a \rangle b]$ $(axiom)$

$\vdash [c, \langle ab \rangle c]$ $(axiom)$

$\vdash [d, \langle abc \rangle d]$ $(axiom)$

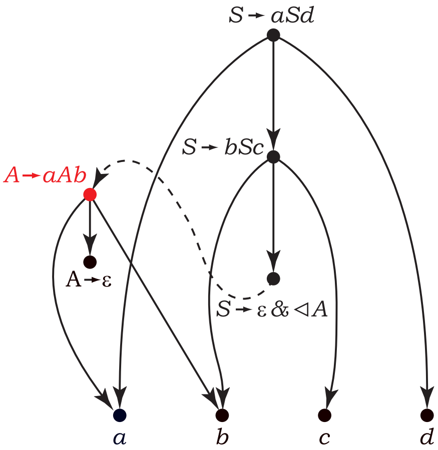
$\vdash [A, \langle a \rangle \varepsilon]$ $(A \rightarrow \varepsilon)$



Example revisited

Grammar for $\{ a^n b^n c^n d^n \mid n \geqslant 0 \}$

$$S \rightarrow aSd \mid bSc \mid \varepsilon \,\&\, \lhd A$$
$$A \rightarrow aAb \mid \varepsilon$$



$\vdash [a, \langle \varepsilon \rangle a]$ (*axiom*)

$\vdash [b, \langle a \rangle b]$ (*axiom*)

$\vdash [c, \langle ab \rangle c]$ (*axiom*)

$\vdash [d, \langle abc \rangle d]$ (*axiom*)

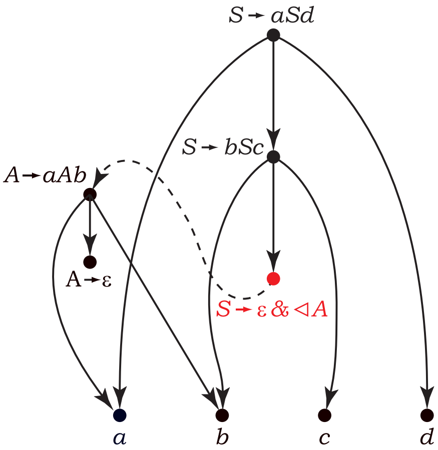
$\vdash [A, \langle a \rangle \varepsilon]$ $(A \rightarrow \varepsilon)$

$[a, \langle \varepsilon \rangle a], [A, \langle a \rangle \varepsilon], [b, \langle a \rangle b] \vdash [A, \langle \varepsilon \rangle ab]$ $(A \rightarrow aAb)$

Example revisited

Grammar for $\{ a^n b^n c^n d^n \mid n \geqslant 0 \}$

$$S \rightarrow aSd \mid bSc \mid \varepsilon \,\&\, \lhd A$$
$$A \rightarrow aAb \mid \varepsilon$$



$\vdash [a, \langle\varepsilon\rangle a]$ $(axiom)$

$\vdash [b, \langle a\rangle b]$ $(axiom)$

$\vdash [c, \langle ab\rangle c]$ $(axiom)$

$\vdash [d, \langle abc\rangle d]$ $(axiom)$

$\vdash [A, \langle a\rangle \varepsilon]$ $(A \rightarrow \varepsilon)$

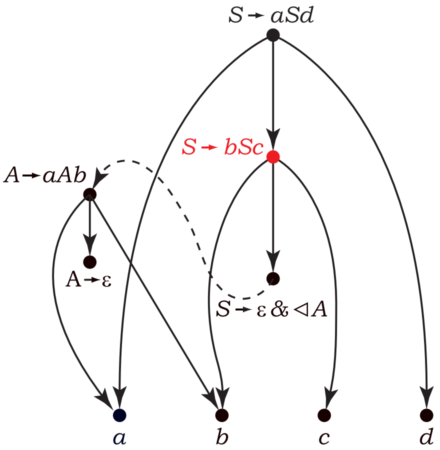
$$[a, \langle\varepsilon\rangle a], [A, \langle a\rangle\varepsilon], [b, \langle a\rangle b] \vdash [A, \langle\varepsilon\rangle ab] \qquad (A \rightarrow aAb)$$

$$[A, \langle\varepsilon\rangle ab] \vdash [S, \langle ab\rangle\varepsilon] \qquad (S \rightarrow \varepsilon \,\&\, \lhd A)$$

Example revisited

Grammar for $\{ a^n b^n c^n d^n \mid n \geqslant 0 \}$

$$S \rightarrow aSd \mid bSc \mid \varepsilon \,\&\, \lhd A$$
$$A \rightarrow aAb \mid \varepsilon$$



$\vdash [a, \langle \varepsilon \rangle a]$ (axiom)

$\vdash [b, \langle a \rangle b]$ (axiom)

$\vdash [c, \langle ab \rangle c]$ (axiom)

$\vdash [d, \langle abc \rangle d]$ (axiom)

$\vdash [A, \langle a \rangle \varepsilon]$ $(A \rightarrow \varepsilon)$

$[a, \langle \varepsilon \rangle a], [A, \langle a \rangle \varepsilon], [b, \langle a \rangle b] \vdash [A, \langle \varepsilon \rangle ab]$ $(A \rightarrow aAb)$

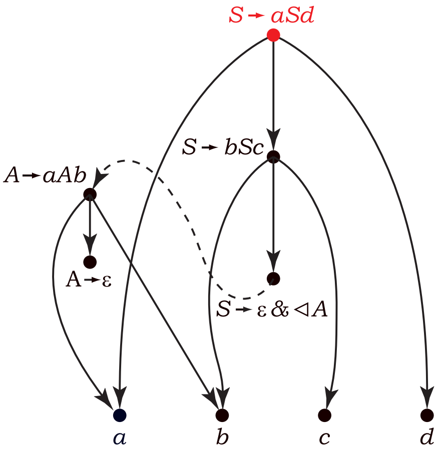
$[A, \langle \varepsilon \rangle ab] \vdash [S, \langle ab \rangle \varepsilon]$ $(S \rightarrow \varepsilon \,\&\, \lhd A)$

$[b, \langle a \rangle b], [S, \langle ab \rangle \varepsilon], [c, \langle ab \rangle c] \vdash [S, \langle a \rangle bc]$ $(S \rightarrow bSc)$

Example revisited

Grammar for $\{ a^n b^n c^n d^n \mid n \geqslant 0 \}$

$$S \to aSd \mid bSc \mid \varepsilon \,\&\, \lhd A$$
$$A \to aAb \mid \varepsilon$$



$\vdash [a, \langle\varepsilon\rangle a]$	$(axiom)$
$\vdash [b, \langle a\rangle b]$	$(axiom)$
$\vdash [c, \langle ab\rangle c]$	$(axiom)$
$\vdash [d, \langle abc\rangle d]$	$(axiom)$
$\vdash [A, \langle a\rangle \varepsilon]$	$(A \to \varepsilon)$

$[a, \langle\varepsilon\rangle a], [A, \langle a\rangle\varepsilon], [b, \langle a\rangle b] \vdash [A, \langle\varepsilon\rangle ab]$	$(A \to aAb)$
$[A, \langle\varepsilon\rangle ab] \vdash [S, \langle ab\rangle\varepsilon]$	$(S \to \varepsilon \,\&\, \lhd A)$
$[b, \langle a\rangle b], [S, \langle ab\rangle\varepsilon], [c, \langle ab\rangle c] \vdash [S, \langle a\rangle bc]$	$(S \to bSc)$
$[a, \langle\varepsilon\rangle a], [S, \langle a\rangle bc], [d, \langle abc\rangle d] \vdash [S, \langle\varepsilon\rangle abcd]$	$(S \to aSd)$

Definition by language equations

- Consider languages of strings in contexts: $L \subseteq \Sigma^* \times \Sigma^*$.

Definition by language equations

- Consider languages of strings in contexts: $L \subseteq \Sigma^* \times \Sigma^*$.

$$K \cup L = \{ \langle u \rangle v \mid \langle u \rangle v \in L \text{ or } \langle u \rangle v \in K \},$$

Definition by language equations

- Consider languages of strings in contexts: $L \subseteq \Sigma^* \times \Sigma^*$.

$$K \cup L = \{ \langle u \rangle v \mid \langle u \rangle v \in L \text{ or } \langle u \rangle v \in K \},$$
$$K \cap L = \{ \langle u \rangle v \mid \langle u \rangle v \in L \text{ and } \langle u \rangle v \in K \},$$

Definition by language equations

- Consider languages of strings in contexts: $L \subseteq \Sigma^* \times \Sigma^*$.

$$K \cup L = \{\, \langle u \rangle v \mid \langle u \rangle v \in L \text{ or } \langle u \rangle v \in K \,\},$$
$$K \cap L = \{\, \langle u \rangle v \mid \langle u \rangle v \in L \text{ and } \langle u \rangle v \in K \,\},$$
$$K \cdot L = \{\, \langle u \rangle vw \mid \langle u \rangle v \in K,\ \langle uv \rangle w \in L \,\},$$

Definition by language equations

- Consider languages of strings in contexts: $L \subseteq \Sigma^* \times \Sigma^*$.

$$K \cup L = \{\, \langle u \rangle v \mid \langle u \rangle v \in L \text{ or } \langle u \rangle v \in K \,\},$$
$$K \cap L = \{\, \langle u \rangle v \mid \langle u \rangle v \in L \text{ and } \langle u \rangle v \in K \,\},$$
$$K \cdot L = \{\, \langle u \rangle vw \mid \langle u \rangle v \in K,\ \langle uv \rangle w \in L \,\},$$
$$\lhd L = \{\, \langle u \rangle v \mid \langle \varepsilon \rangle u \in L,\ v \in \Sigma^* \,\},$$

Definition by language equations

- Consider languages of strings in contexts: $L \subseteq \Sigma^* \times \Sigma^*$.

$$
\begin{aligned}
K \cup L &= \{\, \langle u \rangle v \mid \langle u \rangle v \in L \text{ or } \langle u \rangle v \in K \,\}, \\
K \cap L &= \{\, \langle u \rangle v \mid \langle u \rangle v \in L \text{ and } \langle u \rangle v \in K \,\}, \\
K \cdot L &= \{\, \langle u \rangle vw \mid \langle u \rangle v \in K,\ \langle uv \rangle w \in L \,\}, \\
\lhd L &= \{\, \langle u \rangle v \mid \langle \varepsilon \rangle u \in L,\ v \in \Sigma^* \,\}, \\
\trianglelefteqslant L &= \{\, \langle u \rangle v \mid \langle \varepsilon \rangle w \in L,\ w = uv \,\}.
\end{aligned}
$$

Definition by language equations

- Consider languages of strings in contexts: $L \subseteq \Sigma^* \times \Sigma^*$.

$$\begin{aligned} K \cup L &= \{ \langle u \rangle v \mid \langle u \rangle v \in L \text{ or } \langle u \rangle v \in K \}, \\ K \cap L &= \{ \langle u \rangle v \mid \langle u \rangle v \in L \text{ and } \langle u \rangle v \in K \}, \\ K \cdot L &= \{ \langle u \rangle vw \mid \langle u \rangle v \in K,\ \langle uv \rangle w \in L \}, \\ \lhd L &= \{ \langle u \rangle v \mid \langle \varepsilon \rangle u \in L,\ v \in \Sigma^* \}, \\ \trianglelefteq L &= \{ \langle u \rangle v \mid \langle \varepsilon \rangle w \in L,\ w = uv \}. \end{aligned}$$

$$A = \bigcup_{\substack{A \to \alpha_1 \& \ldots \& \alpha_k \& \\ \& \lhd \beta_1 \& \ldots \& \lhd \beta_m \& \\ \& \trianglelefteq \gamma_1 \& \ldots \& \trianglelefteq \gamma_n \in R}} \left[\bigcap_{i=1}^{k} \alpha_i \cap \bigcap_{i=1}^{m} \lhd \beta_i \cap \bigcap_{i=1}^{n} \trianglelefteq \gamma_i \right] \qquad (A \in N)$$

Definition by language equations

- Consider languages of strings in contexts: $L \subseteq \Sigma^* \times \Sigma^*$.

$$K \cup L = \{\, \langle u \rangle v \mid \langle u \rangle v \in L \text{ or } \langle u \rangle v \in K \,\},$$
$$K \cap L = \{\, \langle u \rangle v \mid \langle u \rangle v \in L \text{ and } \langle u \rangle v \in K \,\},$$
$$K \cdot L = \{\, \langle u \rangle vw \mid \langle u \rangle v \in K,\ \langle uv \rangle w \in L \,\},$$
$$\lhd L = \{\, \langle u \rangle v \mid \langle \varepsilon \rangle u \in L,\ v \in \Sigma^* \,\},$$
$$\trianglelefteq L = \{\, \langle u \rangle v \mid \langle \varepsilon \rangle w \in L,\ w = uv \,\}.$$

$$A = \bigcup_{\substack{A \to \alpha_1 \& \ldots \& \alpha_k \& \\ \& \lhd\beta_1 \& \ldots \& \lhd\beta_m \& \\ \& \trianglelefteq\gamma_1 \& \ldots \& \trianglelefteq\gamma_n \in R}} \left[\bigcap_{i=1}^{k} \alpha_i \cap \bigcap_{i=1}^{m} \lhd\beta_i \cap \bigcap_{i=1}^{n} \trianglelefteq\gamma_i \right] \qquad (A \in N)$$

- Always has a least solution. Let its S-component be $L(G)$.

Definition by language equations

- Consider languages of strings in contexts: $L \subseteq \Sigma^* \times \Sigma^*$.

$$K \cup L = \{ \langle u \rangle v \mid \langle u \rangle v \in L \text{ or } \langle u \rangle v \in K \},$$
$$K \cap L = \{ \langle u \rangle v \mid \langle u \rangle v \in L \text{ and } \langle u \rangle v \in K \},$$
$$K \cdot L = \{ \langle u \rangle vw \mid \langle u \rangle v \in K,\ \langle uv \rangle w \in L \},$$
$$\lhd L = \{ \langle u \rangle v \mid \langle \varepsilon \rangle u \in L,\ v \in \Sigma^* \},$$
$$\trianglelefteq L = \{ \langle u \rangle v \mid \langle \varepsilon \rangle w \in L,\ w = uv \}.$$

$$A = \bigcup_{\substack{A \to \alpha_1 \& \ldots \& \alpha_k \& \\ \& \lhd\beta_1 \& \ldots \& \lhd\beta_m \& \\ \& \trianglelefteq\gamma_1 \& \ldots \& \trianglelefteq\gamma_n \in R}} \Big[\bigcap_{i=1}^{k} \alpha_i \cap \bigcap_{i=1}^{m} \lhd\beta_i \cap \bigcap_{i=1}^{n} \trianglelefteq\gamma_i \Big] \qquad (A \in N)$$

- Always has a least solution. Let its S-component be $L(G)$.

Theorem

The two definitions are equivalent.

Complexity of parsing

Theorem

Every grammar with contexts can be transformed to have rules

$$
\begin{aligned}
A &\to B_1 C_1 \,\&\, \ldots \,\&\, B_k C_k \,\&\, \lhd D_1 \,\&\, \ldots \,\&\, \lhd D_m \,\&\, \trianglelefteq E_1 \,\&\, \ldots \,\&\, \trianglelefteq E_n \\
A &\to a \,\&\, \lhd D_1 \,\&\, \ldots \,\&\, \lhd D_m \,\&\, \trianglelefteq E_1 \,\&\, \ldots \,\&\, \trianglelefteq E_n \\
A &\to B_1 C_1 \,\&\, \ldots \,\&\, B_k C_k \,\&\, \lhd\varepsilon \\
A &\to a \,\&\, \lhd\varepsilon
\end{aligned}
$$

Complexity of parsing

Theorem

Every grammar with contexts can be transformed to have rules

$$\begin{aligned} A &\to B_1 C_1 \,\&\, \dots \,\&\, B_k C_k \,\&\, \lhd D_1 \,\&\, \dots \,\&\, \lhd D_m \,\&\, \trianglelefteq E_1 \,\&\, \dots \,\&\, \trianglelefteq E_n \\ A &\to a \,\&\, \lhd D_1 \,\&\, \dots \,\&\, \lhd D_m \,\&\, \trianglelefteq E_1 \,\&\, \dots \,\&\, \trianglelefteq E_n \\ A &\to B_1 C_1 \,\&\, \dots \,\&\, B_k C_k \,\&\, \lhd \varepsilon \\ A &\to a \,\&\, \lhd \varepsilon \end{aligned}$$

Theorem

Every language generated by a grammar with contexts can be recognized in time $O(|G| \cdot n^3)$.

Complexity of parsing

Theorem

Every grammar with contexts can be transformed to have rules

$$\begin{aligned}
A &\to B_1 C_1 \,\&\, \dots \,\&\, B_k C_k \,\&\, \lhd D_1 \,\&\, \dots \,\&\, \lhd D_m \,\&\, \trianglelefteq E_1 \,\&\, \dots \,\&\, \trianglelefteq E_n \\
A &\to a \,\&\, \lhd D_1 \,\&\, \dots \,\&\, \lhd D_m \,\&\, \trianglelefteq E_1 \,\&\, \dots \,\&\, \trianglelefteq E_n \\
A &\to B_1 C_1 \,\&\, \dots \,\&\, B_k C_k \,\&\, \lhd \varepsilon \\
A &\to a \,\&\, \lhd \varepsilon
\end{aligned}$$

Theorem

Every language generated by a grammar with contexts can be recognized in time $O(|G| \cdot n^3)$.

- cf. $O(|G| \cdot n^\omega)$ for conjunctive and standard context-free grammars.

Complexity of parsing

Theorem

Every grammar with contexts can be transformed to have rules

$$\begin{aligned}
A &\to B_1 C_1 \,\&\, \dots \,\&\, B_k C_k \,\&\, \lhd D_1 \,\&\, \dots \,\&\, \lhd D_m \,\&\, \trianglelefteq E_1 \,\&\, \dots \,\&\, \trianglelefteq E_n \\
A &\to a \,\&\, \lhd D_1 \,\&\, \dots \,\&\, \lhd D_m \,\&\, \trianglelefteq E_1 \,\&\, \dots \,\&\, \trianglelefteq E_n \\
A &\to B_1 C_1 \,\&\, \dots \,\&\, B_k C_k \,\&\, \lhd \varepsilon \\
A &\to a \,\&\, \lhd \varepsilon
\end{aligned}$$

Theorem

Every language generated by a grammar with contexts can be recognized in time $O(|G| \cdot n^3)$.

- cf. $O(|G| \cdot n^\omega)$ for conjunctive and standard context-free grammars.

Theorem

Every language generated by a grammar with contexts is in DSPACE(n).

Conclusions and future work

- "Rule applicable in a context" is indeed an interesting idea.
  - Chomsky just approached it with improper tools.

Conclusions and future work

- "Rule applicable in a context" is indeed an interesting idea.
  - Chomsky just approached it with improper tools.
    - (and thus discovered the first complexity class!)

Conclusions and future work

- "Rule applicable in a context" is indeed an interesting idea.
  - Chomsky just approached it with improper tools.
    - (and thus discovered the first complexity class!)
- Implement further parsing algorithms.

Conclusions and future work

- “Rule applicable in a context” is indeed an interesting idea.
  - Chomsky just approached it with improper tools.
    - (and thus discovered the first complexity class!)
- Implement further parsing algorithms.
  - *Ad hoc* contexts specification is common in recursive descent parsing.

Conclusions and future work

- "Rule applicable in a context" is indeed an interesting idea.
  - Chomsky just approached it with improper tools.
    - (and thus discovered the first complexity class!)
- Implement further parsing algorithms.
  - *Ad hoc* contexts specification is common in recursive descent parsing.
  - $A \to \trianglerighteq B\Sigma^* \& \alpha$

Conclusions and future work

- "Rule applicable in a context" is indeed an interesting idea.
  - Chomsky just approached it with improper tools.
    - (and thus discovered the first complexity class!)
- Implement further parsing algorithms.
  - *Ad hoc* contexts specification is common in recursive descent parsing.
  - $A \to \trianglerighteq B\Sigma^* \& \alpha$
  - Now, a mathematical foundation for that.

Conclusions and future work

- "Rule applicable in a context" is indeed an interesting idea.
  - Chomsky just approached it with improper tools.
    - (and thus discovered the first complexity class!)
- Implement further parsing algorithms.
  - *Ad hoc* contexts specification is common in recursive descent parsing.
  - $A \to \trianglerighteq B\Sigma^* \& \alpha$
  - Now, a mathematical foundation for that.
- Implement two-sided contexts:

$$A \to \alpha_1 \& \dots \& \alpha_k \& \lhd D \& \trianglelefteq E \& \rhd F \& \trianglerighteq G$$

Conclusions and future work

- "Rule applicable in a context" is indeed an interesting idea.
  - Chomsky just approached it with improper tools.
    - (and thus discovered the first complexity class!)
- Implement further parsing algorithms.
  - *Ad hoc* contexts specification is common in recursive descent parsing.
  - $A \to \trianglerighteq B\Sigma^* \& \alpha$
  - Now, a mathematical foundation for that.
- Implement two-sided contexts:

$$A \to \alpha_1 \,\&\, \ldots \,\&\, \alpha_k \,\&\, \lhd D \,\&\, \trianglelefteq E \,\&\, \rhd F \,\&\, \trianglerighteq G$$

- Any negative results?

$$\mathcal{L}(\text{Conj}) \subseteq \mathcal{L}(\text{Conj+contexts}) \subseteq \mathrm{DSPACE}(n) \subseteq \mathrm{NSPACE}(n)$$