Simplification of Context Free Grammars

Same language.
Fewer redundant productions.

ε-productions • Unit productions • Useless symbols

Learning outcomes

Identify

Recognize ε-productions, unit productions and useless symbols.

Transform

Compute nullable variables, unit reachability, generating and reachable variables.

Verify

Preserve the language, including ε when required, through a complete worked example.

Why simplify a grammar?

Goal

Construct an equivalent grammar G′ with L(G′) = L(G).

What changes?

Productions and variables may change; the terminal strings generated must remain the same.

What does not follow?

Simplification does not necessarily remove ambiguity or produce Chomsky Normal Form.

Three kinds of redundant rules

ε-production

A → ε, where ε denotes the empty string.

Unit production

A → B, where both A and B are variables. A → a is not a unit production.

Useless symbol

A variable that cannot participate in a derivation from the start symbol to a terminal string.

Recommended transformation order

1. Remove ε-productions

Introduce a fresh start $S_0 \to S$; preserve $S_0 \to \varepsilon$ if ε belongs to the original language.

2. Remove unit productions

Replace chains of variable-to-variable rules by their non-unit alternatives.

3. Remove useless symbols

First remove non-generating variables; then remove unreachable variables.

Finding nullable variables

Definition

A variable A is nullable if $A \Rightarrow^* \varepsilon$.

Fixed-point algorithm

Start with all A having $A \to \varepsilon$. Add A if $A \to X_1 \ldots X_k$ and every X_i is already nullable.

Example

$A \to \varepsilon$, $B \to A$, $C \to AB$, $D \to aA$. Nullable = {A, B, C}; D is not nullable.

Eliminating ε-productions

Generate alternatives

For each right-hand side, independently omit any subset of occurrences of nullable variables.

Example: S → AB; A → a | ε; B → b | ε

Both A and B are nullable. Introduce $S_0 \to S \mid \varepsilon$. Expand $S \to AB \mid A \mid B$.

Delete ε-rules

$A \to a$; $B \to b$. Keep $S_0 \to \varepsilon$ only; the language remains {ε, a, b, ab}.

Repeated nullable occurrences

Grammar

S → ABA; A → a | ε; B → b.

Omission choices

Keep both A: ABA. Omit first A: BA. Omit last A: AB. Omit both: B.

Result

S → ABA | BA | AB | B; A → a; B → b. Language = {aba, ba, ab, b}.

Eliminating unit productions

Unit reachability

For each A, find all B such that $A \Rightarrow^* B$ using only unit productions; include A itself.

Copy non-unit alternatives

For every reachable B, copy $B \to \alpha$ to A whenever α is not a single variable.

Finish

Delete all unit productions. A protected fresh start may retain its ε-production.

Example: unit chains and cycles

Original grammar

$S \to A \mid b$; $A \to B \mid a$; $B \to A \mid c$.

Unit closures

$U(S) = \{S,A,B\}$; $U(A) = U(B) = \{A,B\}$.

Result

$S \to a \mid b \mid c$; $A \to a \mid c$; $B \to a \mid c$. A and B can later be removed as unreachable.

Removing non-generating variables

Generating variable

A ⇒* w for some terminal string w ∈ Σ* (including ε).

Algorithm

Mark variables with terminal-only right-hand sides; repeatedly mark a variable if all variables on one right-hand side are marked.

Example

S → AB | a; A → a; B → BC; C → c. Generating = {S,A,C}. Delete B and all rules containing B.

Removing unreachable variables

Definition

A is reachable if it occurs in a sentential form derived from the start symbol.

Algorithm

Start with S; repeatedly mark variables appearing on right-hand sides of marked variables.

Example continued

After removing B: S → a; A → a; C → c. Only S is reachable. Final grammar: S → a.

Complete example: initial grammar

Given grammar

$S \to AB \mid C$; $A \to aA \mid \varepsilon$; $B \to b$; $C \to D$; $D \to d$; $X \to x$; $Y \to Y$.

Nullable set

Only A is nullable. S cannot derive ε because B and C are not nullable.

After ε-elimination

$S \to AB \mid B \mid C$; $A \to aA \mid a$; $B \to b$; $C \to D$; $D \to d$; $X \to x$; $Y \to Y$.

Complete example: finish simplification

Remove units

S → AB | b | d; A → aA | a; B → b; C → d; D → d; X → x. Y has no remaining rules.

Remove useless variables

Y is non-generating; C, D and X are unreachable. Retain S, A and B.

Final grammar

S → AB | b | d; A → aA | a; B → b. Language = $\{a^n b : n \geq 0\} \cup \{d\}$.

Classroom exercises

1. ε-elimination

S → AB; A → a | ε; B → b. Remove ε-productions.

2. Unit elimination

S → A; A → B | a; B → b. Remove all unit productions.

3. Useless symbols

S → AB | c; A → a; B → B; C → c. Remove useless variables.

Solutions and recap

1. Nullable A

S → AB | B; A → a; B → b. The unit rule B can be eliminated in the next stage.

2. Unit closures

S → a | b; A → a | b; B → b. A and B are now unreachable.

3. Generating then reachable

B is non-generating; remove S → AB. A and C are then unreachable. Final: S → c.