Context-Sensitive Languages

Teodor Rus

rus@cs.uiowa.edu

The University of Iowa, Department of Computer Science

History

- Context-sensitive grammars and languages were first devised by Noam Chomsky in 1959 as *phrase structure grammars* (PSD), to handle the structure of natural languages.
- Since CFG are inadequate for programming languages, the description of natural languages is not the only a good motivation for more expressive grammars.

That is, context-sensitive grammars are of a larger interest.

Phrase structure grammar

A phrase structure grammar G is a four-tuple $G = (V, \Sigma, R, S)$ where:

1. V and Σ are disjoint, finite, non-empty sets called variables (or non-terminals) and terminals respectively;
2. R is a set of productions (or rules) of the form $lhs \rightarrow rhs$ (sometimes written $lhs = rhs$) where $lhs \in ((V \cup \Sigma)^* \circ V \circ (V \cup \Sigma)^*)$ and $rhs \in (V \cup \Sigma)^*$;
3. $S \in V$ is called start symbol (or axiom).

Note: Since $\epsilon \in (V \cup \Sigma)^*$ regular rules and context-free rules are PSG rules.

Facts

Consider a PSG $G = (V, \Sigma, R, S)$:

1. For each $r \in R$, $lhs(r)$ must always contain at least one variable;
2. No other restriction is required for a pair of strings (lhs, rhs) to qualify as a production of a PSG;
3. Variables of the grammar are meta-symbols, often called *syntax-categories*, whose roles are to aid in the description of languages over Σ^*.

Direct derivation

For a PSG $G = (V, \Sigma, R, S)$ and $x, y \in (V \cup \Sigma)^*$, x **directly derives** y, written $x \Rightarrow_G y$, if:

1. $x = x_1 \alpha x_2$
2. $y = x_1 \beta x_2$
3. $\alpha \rightarrow \beta \in R$

Observations

1. Since β can be ϵ, by a direct derivation we can actually erase a portion of a string;
2. Therefore, PSG rules are also called erasing rules.

Derivation

For a PSG $G = (V, \Sigma, R, S)$ and $x, y \in (V \cup \Sigma)^*$, x **derives** y, written $x \overset{*}{\Rightarrow} y$, iff there exists $z_0, z_1, \ldots, z_n \in (V \cup \Sigma)^*$ such that:

1. $z_0 = x$
2. $z_i \Rightarrow_G z_{i+1}, 0 \leq i < n$
3. $z_n = y$

Notation

- Direct derivation relation is denoted by $\Rightarrow$;
- Derivation relation is denoted by $\stackrel{*}{\Rightarrow}$;
- $x \Rightarrow y$ describes the single application of some production as a rewriting rule to the string x ;
- $x \stackrel{*}{\Rightarrow} y$ denotes the cascading of an arbitrary number of rewriting steps. As a convenience, trivial derivations of zero steps are allowed.

Language generated by a PSG

For a PSG $G = (V, \Sigma, R, S)$, the language generated (or specified) by G is the set $L(G) = \{x \in \Sigma^* | S \stackrel{*}{\Rightarrow} x\}$.

Two PSG G_1 and G_2 are equivalent if $L(G_1) = L(G_2)$

Context-sensitive rule

Let $G = (V, \Sigma, R, S)$ be a PSG.

Definition: A rule $r \in R$, $lhs(r) \rightarrow rhs(r)$, is context-sensitive if $lhs(r) = \alpha A \beta$, $rhs(r) = \alpha\gamma\beta$, $A \in V$, $\alpha, \beta \in (V \cup \Sigma)^*$, and $\gamma \in (V \cup \Sigma)^+$.

Note: because $\gamma \in (V \cup \Sigma)^+$

context-sensitive rules are not erasing.

Grammar hierarchy

Let $G = (V, \Sigma, R, S)$ be a PSG.

- G is a regular grammar if all rules in R are regular, i.e., of the form $A \rightarrow xB|y$ where $A, B \in V$ not necessarily distinct, and $x, y \in \Sigma^*$;
- G is a context-free grammar if all rules in R are context-free, i.e., of the form $A \rightarrow \alpha$, $A \in V$, $\alpha \in (V \cup \Sigma)^*$;
- G is a context-sensitive grammar if all rules in R are context-sensitive, i.e., of the form $\alpha A \beta \rightarrow \alpha \gamma \beta$, $A \in V$, $\alpha, \beta \in (V \cup \Sigma)^*$, and $\gamma \in (V \cup \Sigma)^+$.

Interpretation

- The rewriting of a nonterminal A to γ, with a context-sensitive rule $\alpha A\beta \longrightarrow \alpha\gamma\beta$, requires A to be in the context (α, β);
- Since α and β can be ϵ this generalizes context-free rules;
- Hence, context-sensitive rules provide some control over the rewriting process and thus determine the outcome with more precision.

Chomsky's hierarchy

Let RG, CFG, CSG be the collections of regular, context-free, and context-sensitive grammars, respectively. Then the following relations hold:

- $RG \subseteq CFG \subseteq CSG$
- $RGL \subseteq CFL \subseteq CSL$

Note: This hierarchy is blurred up by CSG requirement $\gamma \in (V \cup \Sigma)^+$ which eliminate the ϵ from CSL while it can be a member of RGL and CFL.

Remedy

- $L \subseteq \Sigma^*$ is an ϵ-**free context-sensitive language** if there exists a CSG G such that $L = L(G)$;
- $L \subseteq \Sigma^*$ is a **context-sensitive language** if there is a context-sensitive grammar G so that $L = L(G)$ or $L = L(G) \cup \{\epsilon\}$, written $L = L_\epsilon(G)$.

Observations

1. Context-sensitive grammar mechanism prohibits the generation of the null string because without erasing, S cannot derive ϵ;
2. Altering the rewriting restriction to permit unrestricted inclusion of ϵ violates the essential character of CSL;
3. The distinction between ϵ-free context sensitive and context sensitive languages is awkward but simplifies the statement of some results;
4. Other authors allow erasing productions of the form $S \to \epsilon$, provided that S appears on the rhs of no rule.

Theorem 2.42

Each context-free language is context-sensitive.

Proof:

1. If L is CFL then there is a CFG G with no erasing rules so that $L(G) = L - \{\epsilon\}$; if $L \neq \emptyset$ and $L \neq \{\epsilon\}$ then G may be chosen to have no useless symbols (see Fleck 211–212).
2. G is context-sensitive and so L is a context-sensitive, i.e., $L = L(G)$ or $L = L_\epsilon(G)$.

Theorem 2.43

Each non-erasing PSG is equivalent to a CSG.

Proof idea: by construction, mapping phrase-structure rules one by one into context-sensitive rules (see Fleck 345-348).

Example 1:

A non-erasing grammar for the language $\{a^{2^k} | k \geq 0\}$ is:

$$G = (\{S, D, A\}, \{a\}, \{S \to DS|A, DA \to aA, A \to a, Da \to aaD\}, S)$$

Sample derivations:

$$S \Rightarrow DS \Rightarrow DDS \Rightarrow DDA \Rightarrow DaA \Rightarrow aaDA \Rightarrow aaaA \Rightarrow a^4$$

$$S \Rightarrow DS \Rightarrow DDS \Rightarrow DDDS \Rightarrow DDDA \Rightarrow DDaA \Rightarrow DaaDA \Rightarrow$$

$$Da^3A \stackrel{*}{\Rightarrow} a^7A \Rightarrow a^8$$

Fact

The following statements can be proved by induction:

- $S \stackrel{*}{\Rightarrow} D^k A$ by context-free rewriting using rightmost derivations;
- Each instance of variable D propagates to the right and acts as a "doubling signal" as it goes;
- Variable A serves as the required termination point, i.e., $D^k A \stackrel{*}{\Rightarrow} a^{2^k - 1} A \Rightarrow a^{2^k}$.

Note: A must be eliminated last or else derivation blocks with no way to eliminate D.

Observation

Derivation process with G cannot progress strictly from left-to-right because none of the doubling passes of D can be completed until after that of those to its right.

Example 2

This example illustrate transformation of the non-erasing grammar in Example 1 into a context-sensitive grammar.

Note: the only non-context sensitive production is $Da \to aaD$. We illustrate the transformation process with this rule.

1. For each $t \in \Sigma$ introduce a new nonterminal X_t. In our case $\Sigma = \{a\}$, hence X_a is the new nonterminal;
2. Add new rules $X_t \to t$ for each nonterminal X_t introduced above;
3. Replace each $t \in \Sigma$ with X_t in the old rules;
4. Transform the non-context-sensitive rules.

Fact

The general technique requires more transformations.

Rule transformation

$DX_a \to X_aX_aD$ is replaced by the rules:

$$\begin{aligned} DX_a &\to Z_1X_a \\ Z_1X_a &\to Z_1Z_2 \\ Z_1Z_2 &\to X_aZ_2 \\ X_aZ_2 &\to X_aX_aD \end{aligned}$$

The equivalent CSG obtained has the rules:

$S \to DS|A, DA \to X_aA, A \to X_a, X_a \to a, DX_a \to Z_1X_a,$
$Z_1X_a \to Z_1Z_2, Z_1Z_2 \to X_aZ_2, X_aZ_2 \to X_aX_aD$

Example 3

A non-erasing grammar for the context-sensitive language $\{a^k b^k c^k | k \geq 1\}$ has productions

$A \to aABC|aBC, aB \to ab, bB \to bb, C \to c, CB \to BC$

with start symbol A.

Note

1. A-rules generate $A \overset{*}{\Rightarrow} a^k(BC)^k$, $k \geq 1$;
2. B and C are not in the right order. The rule $CB \to BC$ sort Bs to the left and Cs to the right;
3. The action performed by the other rules is obvious.

Theorem 2.44

Each phrase structure language is generated by a context-sensitive grammar augmented with context-free erasing rules.

Proof

By construction. Let $G = (V, \Sigma, R, S)$ be a PSG.

1. Choose a new nonterminal $B \notin (V \cup \Sigma)$ and construct $G' = (V \cup \{B\}, \Sigma, R', S)$ where R' consists of all non-erasing rules of R together with $\alpha \rightarrow \beta B^k$, for each erasing rule $\alpha \rightarrow \beta \in R$ and $k = |\alpha| - |\beta|$;
2. Add rule $B \rightarrow \epsilon$ to R';
3. Since B can be erased, it will not interfere with context of other productions; because it does not appear anywhere else in G its presence may be neglected and $L(G) = L(G')$.

Proof, continuation

- Apart from $B \rightarrow \epsilon$, R' contains entirely non-erasing rules;
- By Theorem 2.43, if we remove $B \rightarrow \epsilon$ we obtain an equivalent context-sensitive grammar;
- This grammar together with the rule $B \rightarrow \epsilon$ establish the result.

Theorem 2.45

Each phrase structure language is the homomorphic image of an ϵ-free context-sensitive language.

Proof

By construction. Let $G = (V, \Sigma, R, S)$ be PSG.

1. Choose a new nonterminal B and a new terminal b;
2. Construct $G' = (V \cup \{B\}, \Sigma \cup \{b\}, R', S)$ where R' consists of:
   (a) all non-erasing rules of R;
   (b) rules $\alpha \to \beta^k$, $k = |\alpha| - |\beta|$ for $\alpha \to \beta \in R$, erasing;
   (c) $BX \to XB$ for all $X \in (V \cup \Sigma)$;
   (d) $B \to b$.

Facts

1. $\alpha \overset{*}{\Rightarrow} \beta$ in G iff $\alpha \overset{*}{\Rightarrow} \beta'$ in G' where β' contains β with b-s interleaved;
2. G' has no erasing rules and thus $L(G')$ is a context sensitive language;
3. Define the homomorphism $h : (\Sigma \cup \{b\}) \to \Sigma^*$ by: $h(b) = \epsilon$, $h(w) = w$ for each $w \in \Sigma$, and clearly $h(L(G')) = L(G)$.

Closure property

Definition: A homomorphism $h : \Sigma^* \rightarrow \Delta^*$ is ϵ-free (or non-erasing) if $h(w) \neq \epsilon$ for each $w \in \Sigma$.

Theorem 2.46: context sensitive-languages are closed under ϵ-free homomorphisms.

Proof

Proof: by construction. For $G = (V, \Sigma, R, S)$ a CSG do:

1. Construct $G' = (V \cup \{X_t | t \in \Sigma\}, \Sigma, R' \cup \{X_t \to t\}, S)$ where rules in R' are rules in R where each $t \in \Sigma$ is replaced by X_t;
2. Define $G'' = (V \cup \{X_t | t \in \Sigma\}, \Delta, R' \cup \{X_t \to h(t) | t \in \Sigma\}, S)$;
3. Because h is ϵ-free, G'' is context-sensitives. $S \overset{*}{\Rightarrow} t_1 t_2 \ldots t_n \in \Sigma^*$ in G' iff $S \overset{*}{\Rightarrow} h(t_1)h(t_2) \ldots h(t_n) \in \Delta^*$ in G''. Hence, $L(G'') = h(L(G))$.

Leftmost language

If $G = (V, \Sigma, R, S)$ is a CSG, for $w_1, w_2 \in (V \cup \Sigma)^*$ a rewriting $w_1 \Rightarrow w_2$ is *leftmost*, denoted $w_1 \Rightarrow_L w_2$, if $w_1 = xyA\beta\gamma$, $w_2 = xy\alpha\beta\gamma$, $yA\beta \to y\alpha\beta \in R$, $x, y \in \Sigma^*$.

A derivation is leftmost if each of its step is leftmost.

The leftmost language of G is $LL(G) = \{w \in \Sigma^* | S \overset{*}{\Rightarrow}_L w\}$.

Facts

- $LL(G)$ is considered only for context-sensitive G not for non-erasing G;
- For a CFG G the $LL(G) = L(G)$;
- This is not true for CSG because only productions whose left-context is a terminal string plays a role in the $LL(G)$.

Theorem 2.47

If G is a CSG then $LL(G)$ is context-free.

Proof: by construction. See Fleck 356.