Context Sensitive Grammar

Habeeb and Alvin

ATC Seminar

30 NOV 2018

Overview

Definition

- A context sensitive grammar (CSG) is a grammar where all productions are of the form

$$\alpha A\beta \rightarrow \alpha\gamma\beta \text{ where } \gamma \neq \epsilon$$

- During derivation non-terminal A will be replaced by γ only when it is present in context of α and β.
- This definition shows clearly one aspect of this type of grammar; it is **noncontracting**, in the sense that the length of successive sentential forms can never decrease.

Formal definition

- A context sensitive grammar $G = (N, \Sigma, P, S)$, where
  - N is a set of nonterminal symbols
  - Σ is a set of terminal symbols
  - S is the start symbol, and
  - P is a set of production rules, of the form $\alpha A \beta \rightarrow \alpha \gamma \beta$ where A in N, $\alpha, \beta \in (N \cup \Sigma)^*$ and $\gamma \in (N \cup \Sigma)^+$
- The production $S \rightarrow \epsilon$ is also allowed if S is the start symbol and it does not appear on the right side of any production.

Context Sensitive Language

- A language L is said to be context-sensitive if there exists a context-sensitive grammar G, such that $L = L(G)$.
- If G is a Context Sensitive Grammar then,

$$L(G) = \{w | (w \in \Sigma^*) \wedge \left(S \Rightarrow_G^+ w\right)\}$$

Context Sensitive Language : Example

- Example. The following grammar(G) is context-sensitive

$$S \rightarrow aTb|ab$$
$$aT \rightarrow aaTb|ac$$

$$L(G) = \{ab\} \cup \{a^n cb^n | n > 0\}$$

Context Sensitive Language

$$L(G) = \{ab\} \cup \{a^n cb^n | n > 0\}$$

- This language is also a context-free.
- For example, Context free grammar(G1) for this.

$$S \rightarrow aTb|ab$$

$$T \rightarrow aTb|c$$

- Any context-free language is context sensitive
- Not all context-sensitive languages are context-free.

Context Sensitive Language : Example

Example

$L = \{a^n b^n c^n | n > 0\}$

Context Senstive Language : Example

Example

$L = \{a^n b^n c^n | n > 0\}$

Context sensitive grammar(G)

1. $S \rightarrow aBC$
2. $S \rightarrow \text{a}SBC$
3. $aB \rightarrow ab$
4. $bB \rightarrow bb$
5. $bc \rightarrow bc$
6. $cC \rightarrow cc$
7. $CB \rightarrow CZ$
8. $CZ \rightarrow WZ$
9. $WZ \rightarrow WC$
10. $WC \rightarrow BC$

Closure properties

- Context Sensitive Languages are closed under
  - Union
  - Intersection
  - Complement
  - Concatenation
  - Kleene closure

Closure properties : Union

- Let $G_1 = (N_1, T_1, P_1, S_1)$ and $G_2 = (N_2, T_2, P_2, S_2)$, s.t $L(G_1) = L_1$ and $L(G_2) = L_2$.
- Construct $G = (S \cup N_1 \cup N_2, T_1 \cup T_2, \{S \rightarrow S_1, S \rightarrow S_2\} \cup P_1 \cup P_2, S)$ s.t $N_1 \cap N_2 = \varnothing$ and $S \notin \{N_1 \cup N_2\}$.
- G also CSG and any derivation has the form $S \Rightarrow S_i \Rightarrow^*_{G_i} w \in L(G_i)$ for some $i \in \{1, 2\}$
- We can derive only words and all words of $L(G_1) \cup L(G_2) = L_1 \cup L_2$ Therefore $L_1 \cup L_2 = L(G) \in L(CS)$

Closure properties : Concatenation

- Let $G_1 = (N_1, T_1, P_1, S_1)$ and $G_2 = (N_2, T_2, P_2, S_2)$, s.t $L(G_1) = L_1$ and $L(G_2) = L_2$.
- Construct $G = (S \cup N_1 \cup N_2, T, \{S \rightarrow S_1 S_2\} \cup P_1 \cup P_2, S)$ s.t $N_1 \cap N_2 = \varnothing$ and $S \notin \{N_1 \cup N_2\}$
- Any derivation in G has the form $S \Rightarrow S_1 S_2 \Rightarrow^*_{G_1} w_1 S_2 \Rightarrow^*_{G_2} w_1 w_2$ $S \Rightarrow w_i$ is a derivation in G_i . i.e. the derivation only uses rules of P_i .
- The derivations in G_1 and G_2 cannot be influenced by the contexts of the other part. So G is a context sensitive grammar, $L(G)$ is a CSL.

Relation Between Recursive and CSL

Theorem

Every context-sensitive language L is recursive.

For CSL L, CSG G, Derivation of w $S \Rightarrow x_1 \Rightarrow x_2 \Rightarrow x_3 \cdots \cdot \Rightarrow w$ has bound on no of steps.(Bound on possible derivations). We know that

$$|x_i| \leq |x_{i+1}|$$

(G is non contracting). We can check whether w is in L(G) as follows

- Construct a transition graph whose vertices are the strings of length $\leq |w|$
- Paths correspond to derivation in grammars.
- Add edge from x to y if $x \Rightarrow y$
- $w \in L(G)$ iff there is a path from S to w

References

- An Introduction to Formal Languages and Automata by Peter Linz
- https: / / en. wikipedia. org/wiki/Context-sensitive grammar
- https://gyires.inf.unideb.hu/GyBITT/14/index.html
- old seminars.

The End