Context Sensitive Grammars

Rajaguru K

CSA, IISc

Automata Seminar, Dec,2016

Overview

Definition

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

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

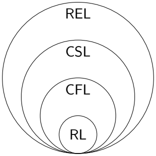
- During derivation non-terminal A will be changed to γ only when it is present in context of α and β.

- Note the constraint that the replacement string $\gamma \neq \epsilon$; as a consequence we have

$$\alpha \Rightarrow \beta \text{ implies } |\alpha| \leq |\beta|$$

- CSG is a Noncontracting grammar.

Chomsky Hierarchy



Type	Language	Automaton	Production rules
Type 0 Unrestricted	Recursively enumerable	Turing Machine	$\alpha \rightarrow \beta$
Type 1 Context-sensitive	Context-sensitive	Linear-bounded automaton	$\alpha A \beta \rightarrow \alpha \gamma \beta$
Type 2 Context-free	Context-free	Pushdown automaton	$A \rightarrow \gamma$
Type 3 Regular	Regular	Finite state automaton	$A \rightarrow a$ *and* $A \rightarrow aB$

Table: Chomsky Hierarchy (1956)

Formal definition

- A context sensitive grammar G = (N, Σ, 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

- The language generated by the Context Sensitive Grammar is called context sensitive language.
- If G is a Context Sensitive Grammar then,

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

Context Sensitive Language

- The language generated by the Context Sensitive Grammar is called context sensitive language.
- If G is a Context Sensitive Grammar then,

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

- 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 but not context-free.

Context Sensitive Language

Example

$L = \{1^n 2^n 3^n \mid n > 0\}$

- Context sensitive grammar(G)

  N = {S, B, C} , Σ = {1, 2, 3} , S = S

  and P:

$$S \rightarrow 1SBC \mid 123$$

Context Sensitive Language

Example

$L = \{1^n 2^n 3^n \mid n > 0\}$

- Context sensitive grammar(G)

  N = {S, B, C} , Σ = {1, 2, 3} , S = S

  and P:

$$S \to 1SBC \mid 123$$
$$1B \to 12$$
$$2B \to 22$$
$$2C \to 23$$
$$3C \to 33$$
$$CB \to BC$$

Context Sensitive Language

Example

$L = \{1^n 2^n 3^n \mid n > 0\}$

- Context sensitive grammar(G)

  N = {S, B, C} , Σ = {1, 2, 3} , S = S

  and P:

$$S \rightarrow 1SBC \mid 123$$
$$1B \rightarrow 12$$
$$2B \rightarrow 22$$
$$2C \rightarrow 23$$
$$3C \rightarrow 33$$
$$CB \rightarrow HB,\ HB \rightarrow HC,\ HC \rightarrow BC$$

Context Sensitive Language

Example

$L = \{x \in \{a, b, c\}^* \mid \#_a x = \#_b x = \#_c x, \ \#_a x \geq 1\}$

- Context sensitive grammar(G)

  N = {S, A, B, C} , Σ = {a, b, c} , S = S

  and P:

$$S \to ABCS \mid ABC$$
$$XY \to YX \text{ for all } X, Y \in \{A, B, C\}$$
$$A \to a$$
$$B \to b$$
$$C \to c$$

Closure properties

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

Closure properties

Union

L(CS) closed under 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 $G = (S \cup N_1 \cup N_2, T_1 \cup T_2, \{S \to S_1, S \to S_2\} \cup P_1 \cup P_2, S)$, 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 cannot merge the productions of P_1 and P_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

L(CS) closed under concatenation

- Let $G_1 = (N_1, T, P_1, S_1)$ and $G_2 = (N_2, T, P_2, S_2)$, 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

$$S \Rightarrow S_1 S_2 \Rightarrow^*_{G_1} w_1 S_2 \Rightarrow^*_{G_2} w_1 w_2$$

  for $i \in \{1, 2\}$, $S_i \Rightarrow w_i$ 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 $S \Rightarrow x_1 \Rightarrow x_2 \Rightarrow x_3 \cdots \Rightarrow w$ 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 thers is a path from S to w.
- Use path fining algorithm to find.

Relation Between Recursive and CSL

Theorem

There exists a recursive language that is not context sensitive.

Language L is recursive

- Create possible CSG $G_i = (N_i, \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, S_i, P_i)$ which generates numbers.
- Now, define language L, which contains the numbers of the grammars which does not generate the number of its position in the list: $L = \{i \mid i \notin L(G_i)\}$.
- We can create a list of all context-sensitive generative grammars which generates numbers, and we can decide whether or not a context-sensitive grammar generates its position in the list.
- So language L is recursive.

Relation Between Recursive and CSL

Theorem

There exists a recursive language that is not context sensitive.

Language L is not context sensitive

- Assume, for contradiction, that L is a CSL
- So there is a CSG G_k, s.t $L(G_k) = L$ for some k.
- If $k \in L(G_k)$, by the definition of L, we have $k \notin L$, but $L = L(G_k)$. So a contradiction.
- If $k \notin L(G_k)$, then $k \in L$ is also a contradiction since $L = L(G_k)$.
- So language L is not context sensitive.

Summary

- Context sensitive languages are closed under union, intersection, complement, concatenation, kleene star, reversal.
- Every Context sensitive language is recursive.
- Further study
  - Proof for other closure properties
  - There is a recursive language that is not context-sensitive.

References

- An Introduction to Formal Languages and Automata by Peter Linz
  Jones & Bartlett Publishers, 2011
- Algebraic Properties of Language Families, by Prof. Dr. Jrgen Dassow
  http://theo.cs.ovgu.de/lehre/lehre09w/ti_1/
- Context Sensitive Grammars
  Klaus Sutner,Carnegie Mellon University, Spring 2016
- Theory of Automata, languages and computation by Rajendra Kumar
  Tata McGraw-Hill, 2010
- and old seminars.