Reduction of PCP into Not Everything Problem for Context Free Languages

Krishna Gummadi

Let { $\frac{w_1}{x_1}, \frac{w_2}{x_2}, ..., \frac{w_k}{x_k}$ } be the collection of dominos for the given PCP problem where lists of strings A $= w_1, w_2, ..., w_k$ & B $= x_1, x_2, ..., x_k \in \Sigma^+$

Let $K = \{a_1, a_2, ..., a_k\}$ be the set of k distinct symbols not in Σ. Define Cfg's G_A, G_B as follows :-

$G_A = (\{S_A\}, V_T, P_A, S_A)$ and $G_B = (\{S_B\}, V_T, P_B, S_B)$ where $V_T = \Sigma \cup K$ and P_A and P_B are defined as follows :-

For All $1 \leq i \leq k$, P_A has rules of the form

$S_A \rightarrow W_i S_A a_i$ and $S_A \rightarrow w_i ai$

For All $1 \leq i \leq k$, P_B has rules of the form

$S_B \rightarrow W_i S_B a_i$ and $S_B \rightarrow w_i ai$

Let $L_A = L(G_A)$ and $L_B = L(G_B)$

Define languages R_{AB} and S_{AB} as $R_{AB} = \{ycy^R / y \in \Sigma^* K^*\}$ and $S_{AB} = \{ycz^R / y \in L_A and L_B\}$. We define $L_{AB} = \bar{R}_{AB} \cup \bar{S}_{AB}$.

Claim:- $L_{AB} = (\Sigma \cup K \cup \{c\})^*$ if and only if PCP with lists A and B has no solution.

Proof:- $L_{AB} = (\Sigma \cup K \cup \{c\})^*$ if and only if $R_{AB} \cap S_{AB} = \phi$. Now, suppose that PCP with lists A and B has a solution. Then there exists a string y in both L_A and L_B and that ycy^R is also in R_{AB} and thus in $R_{AB} \cap S_{AB}$. Conversely, suppose that x is in $R_{AB} \cap S_{AB}$. Then must be of the form ycy^R where y is in both L_A and L_B. Thus PCP with lists A and B has a solution.

It now remains to show that $\bar{R}_{AB}$ and $\bar{S}_{AB}$ are in fact Context free languages that can be generated from the lists. Then it follows from the claim above that the Not Everything problem for context free languages is undecidable as if it were decidable then PCP could be decided too. We show that $\bar{R}_{AB}$ is a cfl by giving the grammar for it, while we show that $\bar{S}_{AB}$ is a cfl by using the properties of deterministic Pda.

We note that Cfl's are closed under union operation while deterministic Pda's are Cfl's which are closed under complementation too.

To show that $\bar{R}_{AB} = \{ycy^R / y \in \Sigma^* K^*\}$ is a cfl:

$\bar{R}_{AB} =$ Set of all strings in $(\Sigma \cup K)^*$ (i.e., strings not containing 'c') — (1)

$\cup$ Set of all strings with a 'c' preceded by a string not in $\Sigma^* K^*$ — (2)

$\cup$ Set of all strings with a 'c' succeeded by a string not in $\Sigma^* K^*$ — (3)

$\cup$ Set of all strings with a 'c' preceded and succeeded by strings of unequal length — (4)

$\cup$ Set of all strings with a 'c' preceded and succeeded by strings of equal length which differ however in the fact that one is not the reverse of the other — (5)

We show that $R_{\bar{AB}}$ is a cfl by showing that (1),(2),(3),(4) and (5) are all cfl's.

(1) $= (\Sigma \cup K \cup \{c\})^* - \Sigma^* K^*$ is clearly a cfl as it is a regular language.

Rules for (2)

$S \rightarrow S_1 c S_2$

where S_1 generates $(\Sigma \cup K \cup \{c\})^* - \Sigma^* K^*$ — is a regular language hence, cfl and therefore, rules can be easily given and

S_2 generates $(\Sigma \cup K \cup \{c\})^*$ — a regular language and hence the rules can be given.

Hence, (2) is a cfl too

(3) can be reasoned out to be a cfl in a manner similar to that of (2).

Rules for (4)

$S \rightarrow aSb$ where $a, b \in (\Sigma \cup K \cup c)$

$\mid S_1 c \mid c S_1$ where S_1 generates $(\Sigma \cup K \cup \{c\})^+$ being a regular language it can be easily given rules.

So, we see that (4) is a cfl too.

Rules for (5)

$S \rightarrow aSb$ where $a, b \in (\Sigma \cup K \cup \{c\})$

$\mid a S_1 b$ where $a \neq b, a, b \in (\Sigma \cup K \cup \{c\})$

$S_1 \rightarrow a S_1 b$ where $a, b \in (\Sigma \cup K \cup \{c\})$

$\mid$ c

So, we see that (5) is a cfl too.

From the above we can conclude that $R_{\bar{AB}}$ is a cfl.

To prove that $S_{\bar{AB}} = \{ycz^R / y \in L_A and L_B\}$ is a cfl:

$S_{\bar{AB}}$ = Set of all strings in $(\Sigma \cup K)^*$ (i.e., not containing 'c') — (6)

$\cup$ Set of all strings containing a 'c' preceded by a string not in y — (7)

$\cup$ Set of all strings containing a 'c' succeeded by a string not in Z^R — (8)

(6) is same as (1). hence, (6) is a cfl.

For (7), we give the production rules below

$S \rightarrow S_1 c S_2$

where S_2 generates $(\Sigma \cup K \cup \{c\})^*$, which being a regular language rules can be easily given.

S_1 generates all strings not of the form $w_{i_1} w_{i_2} ...., w_{i_m} a_{i_m} a_{i_{m-1}} .... a_{i_1} \mid m \geq 1$.

To show that language generated by S_1 is a cfl, we show that the language of the form $w_{i_1} w_{i_2} .... w_{i_m} a_{i_m} a_{im-1} .... a_{i_1} \mid m \geq 1$ can be generated by a deterministic PDA. Since, deterministic PDA's are closed under complementation operation the result would then follow. Since, deterministic PDA's are also closed under 'reverse' operation we show that the reverse of this language i.e., $a_{i_1} a_{i_2} .... a_{i_m} w_{i_m} w_{im-1} .... w_{i_1} \mid m \geq 1$ can be generated by a deterministic PDA. We describe a deterministic PDA which can recognize this language informally below.

The main property of a deterministic PDA is that at any stage of reading the input tape the next state and the stack operation are uniquely determined. So, start on the input tape and as you encounter a_i's just push them onto the stack and proceed till you encounter a symbol in Σ. Once, you encounter a symbol in Σ, pop the corresponding top of the stack a_i and read ahead as long as it matches with the corresponding w_i, if the match fails, reject else if it matches w_i completely, then pop the next symbol from the top of stack and continue.

It can be easily seen that this PDA is deterministic. Hence, it follows that S_1 is a cfl and hence, (7) is a cfl too.

We can reason similarly that (8) is a cfl too.

From above, we can see that $\bar{S_{AB}}$ is a cfl too.

This completes the proof.