Chomsky Normal Form

Definition

Chomsky Normal Form means that every rule in the grammar is either of the form $A \to a$ or $A \to BC$.

Any context free grammar can be converted to a grammar in Chomsky Normal Form (CNF) via a sequence of steps described below.

To follow along either enter the following rules into JFLAP or upload the file CNF.jff.

 $S \rightarrow aAbA$ $S \rightarrow SA$ $A \rightarrow \varepsilon$ $A \rightarrow Y$ $Y \rightarrow bY$ $Y \rightarrow b$

Eliminating epsilon rules

The first step to converting a grammar into CNF is to remove the rules that produce an empty string. In this case there is only one of them $A \to \varepsilon$.

To remove one of these types of rules we basically have to add extra rules that simulate the effect of replacing the variable in question with an ε . For instance $S \to aAbA$, if we replace the first A we get $S \to abA$. replace the second A to get $S \to aAb$ and replace both As with ε to get $S \to ab$. It is important while doing the conversion that every single possibility is added as a rule. We do not want to have any negative effects of eliminating the ε rule.

In JFLAP once you have the grammar loaded up just click on the Transform Grammar option. The first step corresponds to Lambda removal. JFLAP will ask you to click the ε rules and then ask you to remove them one by one. The convenience with JFLAP is that it keeps track of how many rules need to be removed and how many need to be added.

For instance if we just add the $S \rightarrow ab$ rule we get the following message.

Table Tex	t Size					
LHS		RHS		Do Step	Do	All Proceed Export
S S	\rightarrow	aAbA SA		Modify th 1 more r Set that	ie gra emov derive	ammar to remove lambdas. ve(s), and 3 more addition(s) needed. ves lambda: [A]
A	\rightarrow	ε		Delete	Соп	mplete Selected
А	\rightarrow	Y		LHS		RHS
Y	\rightarrow	bY		8	\rightarrow	aAbA
Y	\rightarrow	b		8	\rightarrow	SA
			4	A	\rightarrow	3
		1		A	\rightarrow	Y
				Y	\rightarrow	bY
				Y	\rightarrow	b
			¢.	5	\rightarrow	ab

If you make a mistake you will be informed of it by JFLAP saying it is not part of the new grammar.

and here is the complete picture after eliminating the single ε rule

Table Tex	t Size					
S	\rightarrow	αΔhΔ		Do Step	Do A	Proceed Export
s	\rightarrow	SA		"Proce Set tha	ia rem eed" o at deriv	oval complete. r "Export" available. /es lambda: [A]
Α	\rightarrow	ε		Delet	e Co	mplete Selected
A	\rightarrow	Y		LHS		RHS
Y	\rightarrow	bY	S		\rightarrow	aAbA
Y	\rightarrow	b	S		\rightarrow	SA
	_		A	1	\rightarrow	Y
			Y	7	\rightarrow	bY
			Y	7	\rightarrow	b
			S		\rightarrow	aAb
			S		\rightarrow	ab
			S		\rightarrow	abA
			S		\rightarrow	S

Eliminating unit rules

The next step is eliminating unit rules, rules where one non-terminal just produces another non-terminal. JFLAP makes this easy by basically allowing you to construct a relationship graph.

LHS		RHS	Do Ste	p Do A		Proceed	Expo	rt									
S	\rightarrow	aAbA	Complete	unit prod	ductio	on visual	ization.										
S	\rightarrow	SA	2 more tra	nsition(:	s) nee	eded.			 	 		 				 	
Α	\rightarrow	Y		<u> </u>													
Y	\rightarrow	bY															
Y	\rightarrow	b								<u> </u>							
s	\rightarrow	aAb															
s	\rightarrow	ab															
s	\rightarrow	abA															
s	\rightarrow	S															
															A		
										_							
			(1)														
										s)						
										S)						
			 Automa	on Size					 	 S)					 	
			Automa	on Size					 ▽——	S)					 	
			Automat Delete	on Size	te Sel	ected			 ▽——	 s)	 				 	
			Automa Delete	on Size	te Sel	ected			 Q	 S)	 	RH	S		 	
			Automa Delete	on Size	te Sel	ected			 \	 S)	 	RH	S		 	
			Automa Delete	on Size	te Sel	ected			 ~	 3)		RH	8		 	
			Automar Automar Delete	Complet	te Sel	ected			 \	 8)	 	RH	S		 	
			Automa Delete	complet	te Sel	ected			 Q	S)		RH	 S			
			Automa Delete	Complet	te Sel	ected			 Q	S)	 	RH	<u>s</u>			
			Automa Delete	Complet	te Sel	ected				S)		RH	8			
			Automar Delete	Complet	te Sel	ected			 ~	S)	 	RH	S		 	
			Automar Delete	Complet	te Sel	ected			~	S)		RH	S		 	
			Automar Delete	Complet	te Sel	ected)		RH	8			
			Automa Delete	Complet	te Sel	ected			~)		RH	 S			
			Automa Delete	Complet	te Sel	ected)		RH	8			

Connect up the nodes as per the unit rules.



Removing $A \to Y$. The same idea as removing the ε is used. The effect of having that rule $A \to Y$ is that you get from A to anything that Y can produce. So instead of having to go through this transition we just directly make A go to the things that Y can produce. So extra rules

$$\begin{array}{l} A \to b \\ A \to bY \end{array}$$

have to be added if we want to remove $A \to Y$.

there is still one unit rule $S \to S$ but that can immediately be removed since it is a useless rule.

This results in the following set of rules.

$$S \rightarrow aAbA$$

$$S \rightarrow SA$$

$$Y \rightarrow bY$$

$$Y \rightarrow b$$

$$S \rightarrow aAb$$

$$S \rightarrow abA$$

$$A \rightarrow bY$$

$$A \rightarrow b$$

Adding extra variables

The last step is adding extra variables so that every rule is as per the CNF specifications. This means rules like $S \rightarrow aAbA$ will have to be converted over by adding additional variables. This can be done one at a time from the left as follows.

 $S \to B(a)D(1)$ $D(1) \to AD(2)$ $D(2) \to B(b)A$ $B(a) \to a$ $B(b) \to b$

In general we replace each rule $A \to u_1 u_2 \dots u_k$, where $k \ge 3$ and each u_i is a terminal or non-terminal with the rules $A \to u_1 A_1$, $A_1 \to u_2 A_2$, $A_2 \to u_3 A_3, \dots$, and $A_{k-2} \to u_{k-1} u_k$. The A_i are new variables being introduced. We also need to replace any terminal u_i with $U_i \to u_i$.

The final set of rules look like this

LHS		RHS		Convert	t Selec	ted Do All	What's Left?	Export		
S	\rightarrow	aAbA	All productions completed.							
S	\rightarrow	SA	and the second	LHS						
S	\rightarrow	aAb	10000	S	\rightarrow	\rightarrow B(a)D(1)				
S	\rightarrow	ab	ana ana	D(1)	\rightarrow	AD(2)				
S	\rightarrow	abA		B(a)	\rightarrow	a				
А	\rightarrow	b	1000	B(b)	\rightarrow	b				
А	\rightarrow	bY		s	\rightarrow	SA				
Y	\rightarrow	b		S	\rightarrow	B(a)D(3)	3)			
Y	\rightarrow	bY		D(3)	\rightarrow	AB(b)				
				s	\rightarrow	B(a)B(b)			
				S	\rightarrow	B(a)D(2	2)			
				D(2)	\rightarrow	B(b)A				
				A	\rightarrow	b				
				A	\rightarrow	B(b)Y				
				Y	\rightarrow	b				
				Y	\rightarrow	B(b)Y				