Ling 185A HW 5

Zane Clark

November 2023

3 Relating FSAs to CFGs

D.

For any FSA $(Q, \Sigma, I, F, \Delta)$, one can convert it to a CNF CFG by creating a nonterminal symbol for the start of the FSA path S and one for the start of string character A, a nonterminal symbol, a nonterminal symbol X_i for each node Q_i, a nonterminal symbol T_i for each letter Σ_i. The terminal symbols in the CFG will be the same as the alphabet in the FSA, as well as the start-of-string and end-of-string symbols $\rtimes$ and $\ltimes$, respectively. The starting nonterminal nodes will just be S.

The rules are split into two: for edges (including starting and ending) and for symbols.

The symbol rules are:

- $A \to \rtimes$
- $T_i \to \Sigma_i$

The edge rules are:

- $S \to X_i$ for all initial states Q_i.
- $X_i \to \ltimes$ for all final states Q_i.
- $X_i \to T_j X_k$ for all edges $(Q_i, \Sigma_j Q_k)$

E.

	$\ldots\rtimes$	$\ldots c$	$\ldots v$	$\ldots c$	$\ldots\ltimes$
	S: 0	S: 0	S: 0	S: 0	S: 0
	A: 1	A: 0	A: 0	A: 0	A: 0
	X1: 0	X1: 0	X1: 0	X1: 0	X1: 1
$\rtimes\ldots$	X2: 0	X2: 0	X2: 0	X2: 0	X2: 0
	X3: 0	X3: 0	X3: 0	X3: 0	X3: 1
	C: 0	C: 0	C: 0	C: 0	C: 0
	V: 0	V: 0	V: 0	V: 0	V: 0
		S: 0	S: 0	S: 0	S: 0
		A: 0	A: 0	A: 0	A: 0
		X1: 0	X1: 0	X1: 0	X1: 1
$c\ldots$		X2: 0	X2: 0	X2: 0	X2: 0
		X3: 0	X3: 0	X3: 0	X3: 1
		C: 1	C: 0	C: 0	C: 0
		V: 0	V: 0	V: 0	V: 0
			S: 0	S: 0	S: 0
			A: 0	A: 0	A: 0
			X1: 0	X1: 0	X1: 1
$v\ldots$			X2: 0	X2: 0	X2: 1
			X3: 0	X3: 0	X3: 0
			C: 0	C: 0	C: 0
			V: 1	V: 0	V: 0
				S: 0	S: 0
				A: 0	A: 0
				X1: 0	X1: 0
$c\ldots$				X2: 0	X2: 0
				X3: 0	X3: 1
				C: 1	C: 0
				V: 0	V: 0
					S: 0
					A: 0
					X1: 1
$\ltimes\ldots$					X2: 0
					X3: 0
					C: 0
					V: 0

Some trends in the table are that all of the inside values are 0 for substrings that do not end with $\ltimes$ unless the substring is a single symbol, in which the inside value for the nonterminal node generating that symbol is 1. This makes sense, since all states must end with an end-of-string marker.

The more important pattern is in the last column. These are all the nonterminal nodes which can generate a substring ending with the end of string marker, which is the same as the backwards value.

F.

The rest of the grammar would look like:

$S \to X1\ Z$

$X1 \to \rtimes$

$C \to c$

$V \to v$

$Z \to \ltimes$

$X1 \to \ltimes$

G.

The two CFGs generate the same string, but the one in E does so "forwards" and F does so "in reverse". They both treat states as nonterminal nodes, but E prepends the value caused by each edge and F appends it (and goes the other way for edges).

The top row of the inside table (the substrings starting with $\rtimes$) should look like the forward value table since nonterminal nodes end with the $\rtimes$ symbol in $X1$, so any substring starting with $\rtimes$ has a true inside value for a nonterminal symbol if one can get from the nonterminal signal to the start emitting the substring, which is equivalent to the forward value.