DFAs and CFGs

NFA/DFAs

Symmetric difference: \( A \Delta B = (A - B) \cup (B - A) = (A \cap B) \cup (B \cap \bar{A}) \)

\( A \Delta B \) is the red part (\( A \) is the left circle, \( B \) is the right circle)

NFA to DFA conversion can result in exponential state blow up: \( k \) NFA states \( \rightarrow 2^k \) DFA states

If a \( k \)-state NFA rejects any string, it will have to reject a string of length \( \leq 2^k \), because if you convert the NFA to a DFA and take the complement, you get a DFA for the complement of the NFA’s language. But this is a \( 2^k \) state DFA, which will have to accept a string of length \( \leq 2^k \) if it ever accepts something (\( \iff \) if the NFA ever rejects something).

If a \( k_1 \)-state NFA and a \( k_2 \)-state NFA both accept some string, then the shortest such string has length \( \leq m_1 m_2 \) (because, "we can always remove a segment of the string where a repeated state occurs in both accepting computations of the two NFAs and the number of pairs of states is \( m_1 m_2 \)", Prof. Sipser).

Converting DFAs to regular expressions can kind of blow up in size exponentially. See below:

CFGs

Closure properties

\( CFG \cap \text{REG} \) (intersection with regular languages)

because you can build a PDA that keeps track of the DFA for the regular language and also continues to do the initial PDA’s work.

\( CFG_1 \cup CFG_2 \) (union)

\( CFG^R \) (reversal)

\( CFG_1 \cdot CFG_2 \) (concatenation)

\( CFG^* \) (kleene star)

CFGs are NOT closed under intersection, difference, and complement.

For intersection, consider the following counter-example:

\( A = \{0^m1^n2^n \mid n \geq 0\}, B = \{0^m1^n2^n\} \)
Then $A \cap B = \{0^n 1^n 2^n \mid n \geq 0\}$, which can be proven to be non-context free using the pumping lemma.

**Chomsky normal form**

Chomsky normal form grammars only have productions of the form $A \rightarrow BC \mid \text{terminal}$ and $S \rightarrow \varepsilon$. Thus, any string $w$ in the grammar can be derived in at most $2|w| - 1$ steps.