Homework 5

CSC 445-01: Theory of Computation

Matthew Mabrey, Luke Kurlandski

April 10, 2021

2.15

The construction of G' from CFG G = (V, Σ, R, S) for the CFL A by adding the new rule $S \rightarrow SS$ fails to prove that the class of context-free languages is closed under star because G' may not produce the empty string, ϵ. This is because we do not know if there is a way to produce the empty string from G, in which case we do not know if there is a way to produce the empty string from G'. The Kleene star of a language must always contain ϵ.

2.30

We will prove L is not a context free language using a proof by contradiction and the Pumping Lemma.

$$L = \{t_1\#t_2\#...\#t_k | k \geq 2, t_i \in \{a, b\}^*, \exists(t_i = t_j, i \neq j)\}$$

Suppose L is a CFL. Then for a string $w \in L$ of length greater than p, there exists some decomposition of s, $s = uvxyz$ such that

1. $uv^ixy^iz \in L$ for $i \geq 0$
2. $|vy| > 0$
3. $|vxy| \leq p$

We let $w = a^pb^p\#a^pb^p$ because $|w| \geq p + 1$ and $w \in L$.

1. Since there must be two terms equal to each other and we only have two terms, both sides of # must be equal to each other.
2. Since there must always be at least two terms, v and y cannot include the delimiter # because choosing i = 0 would mean there is no delimiter # and therefore only one term.
3. If v and y were both on one side of the delimiter # then it would fail for every $i \neq 1$ since then $t_1 \neq t_2$. Therefore x must contain the # delimiter and v is to the left of it while y is to the right
4. v must contain only b's and y must contain only a's since if either has both then the other cannot be on the other side of the delimiter # since there are p number of b's before the delimiter # and p number of a's after and $|vxy| <= p$. This remains the case in the situation where v or y is empty as well.
5. Therefore, we are left with the following decomposition:
   - $u = a^pb^{p-q}$
   - $v = b^q$
   - $x = \#$
   - $y = a^r$
   - $z = a^{p-r}b^p$

Its clear that $s = uv^ixy^iz$ results in a string of the form $t_1\#t_2$. But when $i \neq 1$, $t_1 \neq t_2$, thus $s \notin L$. Therefore, we have a contradiction and L is not a context free language.

3.8

b

On input string w

1. Move the head to the front of the tape. Scan the tape and mark the first 1 that has not been marked. If none are found, move to stage 5.
2. Move the head to the front of the tape. Scan the tape and mark the first 0 that has not been marked. If no 0 is found, reject.
3. Scan the tape and mark the first 0 that has not been marked. If no 0 is found, reject.
4. Go to stage 1
5. Move the head to the front of the tape. Scan the tape to see if any unmarked 0s remain. If none remain, accept, else reject.

c

Run the machine from part b on input w. If it accepts, then reject. If it rejects, then accept. Alternatively, build a new TM as described below.

On input string w

1. Move the head to the front of the tape. Scan the tape and mark the first 1 that has not been marked. If none are found, move to stage 5.
2. Move the head to the front of the tape. Scan the tape and mark the first 0 that has not been marked. If no 0 is found, accept.
3. Scan the tape and mark the first 0 that has not been marked. If no 0 is found, accept.
4. Go to stage 1
5. Move the head to the front of the tape. Scan the tape to see if any unmarked 0s remain. If none remain, reject, else accept.

3.15

b

Suppose we have two Turing machines TM_1 and TM_2 that that decide the languages L_1 and L_2 respectively. To determine that the class of decidable languages is closed under concatenation we construct a new machine TM_c that copies the input onto its tape and then inserts a delimiter # at the start of the tape. We then

1. Run TM_1 on all of the input to the left of the delimiter # and TM_2 on all of the input to the right of #
   - If both machines accepts, TM_c accepts the input as part of $L_1 \circ L_2$
   - Else move the delimiter # one position to the right, copying the symbol that was there to #'s old position
   - If we've reached the end of the input and just copied # onto an empty tape symbol, then TM_c rejects the input as not being part of $L_1 \circ L_2$
   - Else go back to step 1

c

Suppose we have a Turing Machine M that decides the language L. We describe the nondeterministic two-tape Turing Machine M' that decides the language L^*.

On input string w

1. Nondeterministically select an integer $k \leq |w|$

2. Nondeterministically partition w into k components, $w_1, w_2, ..., w_k$
3. Run M on each string $w_1, w_2, ..., w_k$
   - If M accepts all the strings $w_1, w_2, ..., w_k$ then accept

d

Suppose we have a Turing Machine M that decides the language L. We describe the Turing Machine M' that decides the language L^c.

On input string w

1. Run M on w.
   - If M accepts, then reject
   - Else accept

e

Having proved that decidable languages are closed under union and complement, DeMorgan's Law proves they are closed under intersection as well (as seen in other classes of languages).

Suppose we have Turing Machines M_1 and M_2 that decide languages L_1 and L_2 respectively. We describe the Turing Machine M' that decides the intersection of L_1 and L_2.

On input string w

1. Run M_1 on w
   - If M_1 accepts, continue to stage 2
   - Else reject
2. Run M_2 on w
   - If M_2 accepts, accept
   - Else reject