CS 2800 HW 6

KIRILL CHERNYSHOV

Problem 1

(a) $n(ZD_1D_1D_0) = 2n(ZD_1D_1) = 2(2n(ZD_1)+1) = 2(2(2n(Z)+1)+1) = 2(2(1)+1) = 2(2+1) = 2(3) = 6$.

(b) Define $\text{incr} : B \to B$ such that $\text{incr} : Z \mapsto ZD_1, bD_0 \mapsto bD_1, bD_1 \mapsto \text{incr}(b)D_0$.

(c) Let $P(a)$ be the statement that $n(\text{incr}(a)) = n(a)+1$. The base case is $P(Z)$, which is trivial to prove: $n(\text{incr}(Z)) = n(ZD_1) = 2n(Z)+1 = 1 = n(Z)+1$. For the inductive step, we need to assume $P(b)$ holds, and show that this implies $P(bD_0)$ and $P(bD_1)$.

Proof. For the first case, $n(\text{incr}(bD_0)) = n(bD_1) = 2n(b)+1 = n(bD_0)+1$. For the second case, $n(\text{incr}(bD_1)) = n(\text{incr}(b)D_0) = 2n(\text{incr}(b)) = 2(n(b)+1) = 2n(b)+2$. Since $n(bD_1) \equiv 2n(b)+1$, we have $n(b) = \frac{n(bD_1)-1}{2} \implies 2n(b)+2 = n(bD_1)-1+2 = n(bD_1)+1$. Therefore, $P(b) \implies P(bD_0), P(bD_1)$. □

Therefore, this statement is true.

(d) Define $\text{add} : B \times B \to B$ as a symmetric function ($\text{add}(a,b) = \text{add}(b,a)\ \forall\ a,b \in B$), with values as follows:

a	b	$\text{add}(a,b)$
Z	b	b
aD_0	bD_0	$\text{add}(a,b)D_0$
aD_0	bD_1	$\text{add}(a,b)D_1$
aD_1	bD_1	$\text{incr}(\text{add}(a,b))D_0$

(e) We can prove the distinct cases one by one. Firstly, $n(\text{add}(Z,b)) = n(b) = n(b)+0 = n(b)+n(Z)$.

Then, let $P(p,q)$ be the statement that $n(\text{add}(p,q)) = n(p)+n(q)$. The base cases are: $P(Z,b)$ and $P(b,Z)$, which we know to be true per case 1. The inductive step is showing that $P(p,q) \implies P(pD_0,q), P(p,qD_0), P(pD_1,q), P(p,qD_1), P(pD_0,qD_0), P(pD_0,qD_1), P(pD_1,qD_0), P(pD_1,qD_1)$. $P(pD_0,q) \iff P(p,qD_0)$ and $P(pD_1,q) \iff P(p,qD_1)$ because the add function is symmetric, as is the function $a,b \mapsto n(a)+n(b)$. Also, $P(Z,b), P(aD_0,bD_0), P(aD_0,bD_1) \implies P(aD_0,b)$ since b must either be Z or of the form kD_0 or kD_1 for some $k \in B$, which means $P(aD_0,b) \iff P(aD_0,kD_0), P(aD_0,kD_1)$. Similarly, $P(Z,b), P(aD_1,bD_0), P(aD_1,bD_1) \implies P(aD_1,b)$. Therefore, the only other cases we need to prove are $P(aD_0,bD_0)$, $P(aD_0,bD_1)$ and $P(aD_1,bD_1)$, with the assumption of $P(a,b)$.

Proof. $P(aD_0,bD_0)$: $n(\text{add}(aD_0,bD_0)) = n(\text{add}(a,b)D_0) = 2n(\text{add}(a,b)) = 2(n(a)+n(b)) = 2\left(\frac{n(aD_0)}{2}+\frac{n(bD_0)}{2}\right) = n(aD_0)+n(bD_0)$.

$P(aD_0,bD_1)$: $n(\text{add}(aD_0,bD_1)) = n(\text{add}(a,b)D_1) = 2n(\text{add}(a,b))+1 = n(aD_0)+n(bD_0)+1 = n(aD_0)+(n(bD_1)-1)+1 = n(aD_0)+n(bD_1)$.

Finally, $P(aD_1,bD_1)$: $n(\text{add}(aD_1,bD_1)) = n(\text{incr}(\text{add}(a,b))D_0) = 2n(\text{incr}(\text{add}(a,b)))$. Since $n(\text{add}(a,b)) = n(a)+n(b)$ by $P(a,b)$, and $n(\text{incr}(k)) = n(k)+1\ \forall\ k \in B$, we have $n(\text{incr}(\text{add}(a,b))) =$

$n(a) + n(b) + 1$, and therefore $2n(\text{incr}(\text{add}(a, b))) = 2(n(a) + n(b) + 1) = 2n(a) + 2n(b) + 2 = (2n(a) + 1) + (2n(b) + 1) = n(aD_1) + n(bD_1)$. $\square$

Therefore, $P(a, b)$ is true for all $a \in B, b \in B$.

Problem 2 The alphabet for all three machines is $\Sigma = \{0, 1\}$. Assume q_0 is always the starting state.

(a) For this we need five states: q_0, q_1, q_2, q_3, q_4. The accepting states are $q_i : 0 \leq i \leq 3$. The transition function δ is as follows:

q	σ	$\delta(q, \sigma)$
q_0	0	q_1
q_1	0	q_2
q_2	0	q_3
q_3	0	q_4

with $\delta(q_i, 1) = q_i \; \forall \; i$ and $\delta(q_4, i) = q_4 \; \forall \; i$.

In this case, a string with no 0s will end up where it started, in state q_0. A string with one 0 will end up in q_1, a string with two zeroes will end up in q_2 and one with three will end up in q_3. All of these are accepting states. A further 0 will push a string into q_4 where it will stay from then on regardless of further letters, because any string with more than three 0s is to be rejected no matter what. The transition function does not change state in case of a 1, since 1s have no effect on whether a string has more or less than three 0s.

(b) For this we need five states: q_0, q_1, q_2, q_3, q_4. The accepting state is q_3. The transition function δ is as follows:

q	σ	$\delta(q, \sigma)$
q_0	0	q_1
q_0	1	q_3
q_1	0	q_3
q_1	1	q_2
q_2	0	q_3
q_2	1	q_3

Furthermore, $\delta(q_3, i) = q_4, \delta(q_4, i) = q_4 \; \forall \; i$.

A string will start out in q_0, where it does not satisfy the condition yet. If a 1 is received, then the string now satisfies the condition as it is 1, one of the four accepted strings, so it goes to q_3 where it is accepted. If a 0 is received, the string goes to q_1. Then, if there is a 0 the string goes to q_3, as it is now 00, one of the four accepted strings. If however the next letter is 1 the string goes to q_2. In q_2, whatever letter is received next the string goes to q_3, as 010 and 011 are both accepted strings.

If a string is in q_3 and any other letter is received, it goes to q_4, where it stays, since a string in q_4 will stay there regardless of further letters. This is because appending either 0 or 1 to any of the four accepted strings makes it a non-accepted string, such that no further additions could make it accepted. Appending to 010 and 011 would make a 4 letter string, all of which are rejected. Appending to 1 means the string must always be rejected, as no accepted string with more than one letter starts with a 1. Finally, for the same reason appending anything to 00 means the string must always be rejected. Thus, any accepted string that receives another letter must be rejected with no possibility of further acceptance.

(c) For this we need three states: q_0, q_1, q_2. The accepting state is the starting state q_0. The transition function δ is as follows:

q	σ	$\delta(q, \sigma)$
q_0	0	q_0
q_0	1	q_1
q_1	0	q_0
q_1	1	q_1
q_2	0	q_2
q_2	1	q_2

A string will start out in q_0, which is an accepting state. This is because the empty string satisfies the condition. Any 0s will keep the string in q_0 as it would still satisfy the condition. If there is a 1 it moves to q_1, and it does not satisfy the condition anymore. If the next letter is also 1 then the string will move to q_2 where it will stay regardless of further letters, since it has already broken the condition of any 1s being followed by 0s. However if the next letter is a 0, the string once again satisfies the condition, and returns to q_0.