ITCS 532:
W6 Homework Solutions

Rob Egrot

Q1

Prove that $\equiv_p$ is an equivalence relation.

- We know $\leq_p$ is reflexive by exercise 7.5.1 in the notes, so clearly $A \equiv_p A$.
- We also know $\leq_p$ is transitive by theorem 7.5.2.
- So if $A \equiv_p B$ and $B \equiv_p C$, then by definition of $\equiv_p$ we have $A \leq_p B$ and $B \leq_p C$, so $A \leq_p C$.
- Similarly we have $C \leq_p A$ and so $A \equiv_p C$.
- So $\equiv_p$ is transitive.
- Finally, $A \equiv_p B \iff A \leq_p B$ and $B \leq_p A$, and so $\equiv_p$ is obviously symmetric.

Q2

Let $k \in \mathbb{N} \setminus \{0\}$.

a) Prove that k^{n+1} is $O(k^n)$ as a function of n.

b) Prove that k^{2n} is not $O(k^n)$ as a function of n.

a)
- We have $k^{n+1} = k.k^n$ for all n.
- So $k^{n+1} \leq ck^n$ when $c = k$. I.e. k^{n+1} is $O(k^n)$.

b)
- Suppose k^{2n} is $O(k^n)$.
- Then there is a constant c with $k^{2n} \leq ck^n$ for 'large' n.
- Taking $\log_k$ of both sides gives

$$2n \leq \log_k c + n,$$

and so

$$n \leq \log_k c,$$

but this obviously cannot be true for all 'large' n.

Q3

Let C_1 and C_2 be computers, and suppose that C_2 is 2^9 times faster than C_1. I.e. if C_1 can take v_1 computation steps per second then C_2 can take $v_2 = 2^9 v_1$ computation steps per second. C_1 and C_2 both run the same $O(n^3)$ algorithm for sorting a list of numbers of size n. Suppose the largest list size that C_1 can guarantee to sort in some fixed time t is 1000. I.e. C_1 can guarantee to sort a list with size at most 1000 in time t (a whole number of seconds), but larger lists may take longer. Approximately what is the largest list size that C_2 is guaranteed to sort in the same time t?

Q3

- Let n_i be the largest list size that C_i is guaranteed to sort within time t ($i \in \{1, 2\}$).
- Since the algorithm is $O(n^3)$, there is a constant c such that the run time on a list of length n is at most cn^3.
- Number of steps C_i can take in time t is $v_i t$.
- So $n_i = \max\{n : cn^3 \leq v_i t\}$.
- I.e. $n_1 = \lfloor (\frac{v_1 t}{c})^{\frac{1}{3}} \rfloor$, and $n_2 = \lfloor (\frac{v_2 t}{c})^{\frac{1}{3}} \rfloor$.
- But $v_2 = 2^9 v_1$, so $n_2 = \lfloor (\frac{2^9 v_1 t}{c})^{\frac{1}{3}} \rfloor \approx 2^3 \lfloor (\frac{v_1 t}{c})^{\frac{1}{3}} \rfloor = 2^3 n_1$.
- We know $n_1 = 1000$, so $n_2 \approx 2^3 \times 1000 = 8000$.

Q4

Let $\Sigma = \{0, 1\}$. Let D be the decision problem "Given a Turing machine T over Σ, does $T(I)$ halt whenever $|I|$ is odd?".

a) Define the empty tape halting problem ($ETHP$).

b) Show that $ETHP \leq D$.

c) Is D decidable? Justify your answer.

Q4

a) "Given a Turing machine T, does T halt when run on a blank tape?".

b)
- An instance of $ETHP$ is a Turing machine T.
- An instance of D is also a Turing machine.
- Given a Turing machine T we will construct T' such that T halts on the empty input iff $T'(I)$ halts whenever $|I|$ is odd.
- T' operates on an input I as follows:
  1) First T' erases I and moves the tape head back to the start of the tape.
  2) T' then does what T do (note the tape is now empty).
- Then $T(\epsilon)$ halts implies $T'(I)$ halts for all I, in particular whenever $|I|$ is odd.
- Conversely, T' does the same thing for all odd length inputs, and $T'(I)$ halts for all I with $|I|$ odd implies $T(\epsilon)$ halts.

c) $ETHP \leq D$, so D is undecidable.