CSE 431
Spring Quarter 2001
Assignment 7
Due Friday, May 25

All solutions should be neatly written or type set. All major steps in proofs and algorithms must be justified.

1. (10 points) The following algorithm solves the Subset Sum problem in pseudo polynomial time. With input `a[1],...,a[n],b` do the following. Let `X` be an array of Boolean values with range `0` to `b`.

```
for j = 1 to b do X[j] = 0;
X[0] = 1;
for i = 1 to n do
  for j = b to 0 by -1 do
    if X[j] == 1 and j + a[i] <= b then
      X[j + a[i]] = 1;
return X[b]
```

   Unfortunately, this algorithm only determines whether or not there is a solution. Use this algorithm as a basis of an algorithm to solve the problem of finding a solution to the Subset Sum problem if there is one. Your algorithm should run in time $O(bn)$.

2. (10 points) Define a $\{\cup, \cdot\}$-regular expression as one that only uses union and concatenation, but not Kleene star. We define the Not Everything problem for $\{\cup, \cdot\}$-regular expressions by:

   Input: A $\{\cup, \cdot\}$-regular expression α over Σ and number k.
   Property: $L(\alpha) \neq \Sigma^k$.

   Show that the Not Everything Problem for $\{\cup, \cdot\}$-Regular Expressions is NP-complete.

   Note that showing the problem is in NP is not altogether trivial. A first step would be to convert α into an NFA. I would recommend showing that 3-CNF-SAT is mapping reducible in polynomial time to this problem. If the CNF formula F uses Boolean variables $x_1, x_2, \ldots, x_n$ then a string in $\{0,1\}^n$ can represent an assignment to the variables. Try to design a $\{\cup, \cdot\}$-regular expression α over the alphabet $\{0,1\}$ with the property that $w \in L(\alpha)$ if and only if w is of length n that does not represent a satisfying assignment of F. In other words, $L(\alpha) \neq \{0,1\}^n$ if and only if F is satisfiable.