CS 172, Spring 1999
Midterm #2 Solutions
Professor Manuel Blum

Problem #1

a) 1011*1101 = 10001111 b) yes yes yes no no The n-bit numbers are all x in Z^+ 2^(n-1) <= x < 2^n The product of an m-bit number x and an n-bit number y is therefore x*y with 2^(m+n-2) <= x*y < 2^(m+n). So The product of an m-bit by n-bit number, for m, n >= 1, is either m+n or m+n-1 bits.

Problem #2

YES: (1+x)^2 + y^4(1+z^6) 3 = 0 x = 0 y = 1 z = 1

Problem #3

Don't know! However, if that's your complete answer, you won't get much credit. Let's see what's going on: Part 1: Note: to be in NP, the length of the proof (NP proof) must be upper bounded by a fixed polynomial of the input length (length of f). [i.e. the time to compute f(x_1, x_k) for given values of x_1, , x_k in {0, 1} must be <= poly(length of f(x_1, , x_k)) for some fixed polynomial.] Part 2: If the definition of multivariate polynomial did not permit any exponentiation at all (so that x^3 would have to be written x*x*x), then the answer would be YES. That's because + and * cannot increase the length of an expression: an m-bit + n-bit number has length <= 1 + MAX{m, n} An m-bit * n-bit number has length <= m+n Part 3: If the definition of multivariate poly did permit raising to a unary power, but did not permit nesting powers (such as ((2^11)^11)^11 = ((2^2)^2)^2 = 2^(2*2*2) = 2^(2^3)), then the answer would be YES. That's because the length of an (m-bit binary number)^((n-m)-bit unary power) à is (m + + m)= m(n-m), which is maximized at m = n/2. Therefore max = n^2/4. In this case, the fixed polynomial would be O(n^2). Part 4: But we are allowing nested powers. So we can get big numbers; eg. ((2^11)^11)^11 = 2^(2*2 2) = 2^(2^k), which has length 2^k. This does not imply that ZMP is NOT in NP, since it might be possible to get the final answer to the decision problem (YES or NO) without generating such large numbers. For example, might succeed by doing all computations modulo small primes. =GOOD PHD PROBLEM!=

Problem #4

Don't know: Same reasoning as problem 3, though the final answer, when known, might be different.

Problem #5

YES. Transform by mapping. 3SAT à ZMP T à 0 F à 1 + (OR) à * * (AND) à + X à x^2 Conj(X) à (1-x)^2 Ex: (A + B + conj(C))(B + C)(conj(B)) à a^2 b^2 (1-c)^2 + b^2 c^2 + (1-b)^2 The map is poly-time

Problem #6

YES. Follows from part 5. Transform ZMP with degrees in unary à ZMP with degrees in ginary Ex: x^(1111111111111) à x^1101 In general, the map takes a multivariate poly expression into a shorter (or at least not longer) expression. It's poly-time. It maps YES à YES NO à NO
