University of Copenhagen

Assignment 2: Counting parentheses

Jens Fredskov (chw752)

September 29, 2013

1 Introduction

The following report describes the implementation and testing of Piano numbers, different arithmetic operations and prime number operations in Prolog.

2 Implementation

The library has been implemented in the file `nats.pl`. The file implements comparison (only less than), addition, multiplication and modulus.

We could further implement minus, power and square root using these. Minus could be done by simply querying for the first or second argument of add, since this corresponds to either $n_3 - n_1 = n_2$ or $n_3 - n_2 = n_1$. Power could be implemented using multiplication where we multiply the argument with itself as many times as needed. As with minus square root could be implemented by querying the multiplication for a known result with the multiplicands being the same argument, e.g $mult(X, X, 9)$.

The file also implements predicates to query for assert whether a number is a prime, list a number of primes, and assert whether a number is a super-prime.

3 Testing

The predicates have been tested using the test predicates in `nats.pl` which allows us to use normal base 10 numbers, allowing for easier testing.

These have been used to test the predicates for randomly selected numbers. All of the functions have worked as expected when tested.

The *listPrimes* predicate with numbers greater than 12 yields several results, which is undesirable. All of them are however the same, and correct. The exact reason is not clear, but it seems to be due to the way the predicate chooses between the first or second clause.

When using using *mult* with the example from the assignment text it returns the correct output and halts. The query halts because addition is implemented in such a way that it will only give one result and because the first two arguments are bounded using *less*.

When querying `notprime(X)`, the following is obtained:

$$[0, 1, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, \ldots]$$

The first two because they are handled by facts, whereas the rest is all even numbers.

When querying `prime(X)` no non-primes are shown up at least up to 100.

4 Conclusion

We have now described the implementation. The testing of the implementation has been described, and we have concluded that the implemented parts of the library has worked as expected in all of our tests.