Rule-based Programming

Assignment #4

Excercise 1 (Top-Down Fibonacci)**.** Consider the following two top-down implementations of Fibonacci numbers.

Variant 1:

```
fib(0,M) <=> M=1.
fib(1,M) <=> M=1.
fib(N,M) <=> N>=2 | N1 is N-1,
                    N2 is N-2,
                    fib(N1,M1),
                    fib(N2,M2),
                    M is M1+M2.
```

Variant 2:

```
fib(N,M1) \ fib(N,M2) <=> M1=M2.
fib(0,M) ==> M=1.
fib(1,M) ==> M=1.
fib(N,M) ==> N>=2 | N1 is N-1,
                    N2 is N-2,
                    fib(N1,M1),
                    fib(N2,M2),
                    M is M1+M2.
```

Predict the time complexity of each variant. Which variant is more efficient? Implement both variants and verify your prediction.

Excercise 2 (Bottom-Up Fibonacci)**.** Implement a bottom-up implementation of Fibonacci numbers based on the following rule (which is non-terminating in the current form):

```
fib(N1,M1), fib(N2,M2) ==> N2=:=N1+1 | N3 is N2+1,
                                       M3 is M1+M2,
                                       fib(N3,M3).
```

How do you ensure termination?

Excercise 3 (Top-Down Factorial)**.** Implement a CHR program computing the factorial of a natural number in a top-down manner.

Excercise 4 (Top-Down Binomial Coefficients)**.** Binomial coefficients (*Pascal's triangle*) can be computed using the following recursive formula:

$$\binom{n}{k} ::= \begin{cases} 1 & \text{if } k = 0 \text{ or } k = n \\ \binom{n-1}{k} + \binom{n-1}{k-1} & \text{if } 0 < k < n \\ 0 & \text{otherwise} \end{cases}$$

Implement a CHR program to compute binomial coefficients efficiently in a top-down manner.

Excercise 5 (Bottom-Up Binomial Coefficients)**.** Implement a CHR program to compute binomial coefficients in a bottom-up manner.

Excercise 6 (Fibonacci Approximation of the Golden Ratio)**.** The so-called "golden ratio" (*sectio aurea* in latin) has been known for at least 2.400 years and among other properties has been widely considered as aesthetically especially pleasing. Mathematically, the golden ratio is an irrational constant which equals

$$\varphi = \frac{1 + \sqrt{5}}{2}$$

The golden ration is closely related to the Fibonacci sequence:

$$\lim_{n \to \infty} \frac{F(n+1)}{F(n)} = \varphi$$

Use this relationship to implement a CHR program which computes an approximation to the golden ratio up to some user-defined accuracy.