Constructive Mathematics (part I)

Conceive of proof as *algorithmic*.
(So instead of true/false, we distinguish provable/provably unprovable/unknown.)

The BHK (Brouwer Heyting Kolmogorov) interpretation:

CLASS		BHK
P is true.	P	We have a proof of P.
Both P and Q are true.	$P \wedge Q$	We have a proof of P and a proof of Q.
P is true or Q is true.	$P \vee Q$	We have either a proof of P or a proof of Q.
Q is true whenever P is true.	$P \to Q$	We have an algorithm that converts proofs of P into proofs of Q.
P is not true.	$\neg P$	We have a proof that $P \to \bot$ ($\bot$ is absurd).
P holds for each $x \in A$.	$\forall_{x\in A} P(x)$	We have an algorithm which converts any element $x \in A$ into a proof of $P(x)$.
Some x in A is such that $P(x)$.	$\exists_{x\in A} P(x)$	We have an algorithm which computes an object $x \in A$ and confirms that $P(x)$.

Adopting the BHK interpretation makes a lot of sense computationally, because constructive proofs:

(a) embody (in principle) an algorithm (for computing objects, converting other algorithms, etc.), and

(b) prove that the algorithm they embody is correct (i.e. that it meets its design specification).

The downside is that we lose some powerful tools in our arsenal, such as

- the Law of Excluded Middle (LEM); for any proposition P either P or $\neg P$,
- careless use of proof by contradiction,
- unrestricted double negation elimination. ($\neg\neg P \to P$ for arbitrary P)

Example

Why do we lose LEM? Given a formal theory T, consider a sentence reminiscent of Gödel's theorem:

> G: G is not provable in T.

Under the BHK interpretation, LEM applied to G (within the theory T) gives:

> We have a proof of G, or a proof that $G \to \bot$.

But now if we have a proof of G, we prove something false (and our system is inconsistent); or if we have a proof that $G \to \bot$ then we have *proven* that G is unprovable (which, if we read G carefully, is a proof of G)! So we may assert LEM here only on pain of inconsistency.

This is not to say that LEM is false; merely that there is no *algorithmic means* for *deciding* whether P is true or false for arbitrary P.

Incidentally, this means we also lose anything which validates LEM, such as double negation elimination ($\neg\neg P \to P$ is not, in general, algorithmically provable for arbitrary P). So we cannot, in general, prove $\exists_x P(x)$ by assuming $\neg\exists_x P(x)$ and deriving a contradiction; it doesn't *compute* the required x.

Example

Consider the classical least upper bound principle (LUB), a cornerstone theorem of classical analysis:

> Every nonempty set S of real numbers that is bounded above has a least upper bound.

It should be pretty clear that there need not be an algorithm which computes this supposed least upper bound. In fact, it is an *essentially nonconstructive* principle, as this example shows. Consider the set

$$S = \{x \in \mathbb{R} : (x = 2) \vee (x = 3 \wedge P)\},$$

where P is some as-yet undecided (unproven) proposition, such as the Goldbach conjecture (every even integer greater than 2 is the sum of two primes).
S is nonempty ($2 \in S$) and bounded above (by 4, for example). However, suppose that we can compute its least upper bound. Then either its l.u.b. is greater than 2 (in which case $3 \in S$ and so P must be true), or its l.u.b. is less than 3 (in which case P will lead to a contradiction, and so $\neg P$).
Therefore we have proven

> If every nonemty set S of real numbers that is bounded above has a least upper bound, then LEM holds.

This is a so-called *Brouwerian example* (or *weak counterexample*), which shows that the constructive interpretation of a classical theorem implies some non-constructive principle; and therefore cannot be constructively *proved.* (Note that this does not imply that it is *false.*)

Compare to this the *constructive* least-upper-bound principle:

> An inhabited set S of real numbers that is bounded above *and is order located* has a least upper bound.

A set is (upper) order located if for each $a, b \in \mathbb{R}$ with $a < b$ either b is an upper bound for S or there exists (we can compute) $x \in S$ with $a < x$.

Moral: in constructive mathematics, computational *information* is of utmost importance; the order-locatedness of a set provides computational information on how close you can get to its least upper bound (i.e. arbitrarily close).