Two philosophies – two logics
Dirk van Dalen (HB of philosophical logics):

Among the logics that deal with the familiar connectives and quantifiers two stand out as having a solid philosophical-mathematical justification. On the one hand there is classical logic with its ontological basis, and on the other hand there is intuitionistic logic with its epistemic motivation.

- ontological basis of **CL**: Valid means ‘invariantly true?’
  → concept of ‘platonic truth’ (modelled by Tarski semantics)

- epistemic motivation of **IL**: valid means ‘invariantly known?’
  → concept of ‘ideal knowledge’

‘invariantly’ refers to all interpretations of non-logical symbols

Warning: at the propositional and at the first-order logic level **IL** is a subset (‘fragment’) of **CL**. However, at higher orders they are incompatible: Intuitionistic (and constructive) mathematics is very different from classical mathematics!

However for finite domains (theories in) **IL** and **CL** coincide.
Tertium non datur !??

What’s the relevance for computer science?

Either the following program $\pi$ terminates or it doesn’t terminate:

**INPUT:** $n$ of type *positive integer* (unbounded)
while $n > 1$ do
  if even($n$) then $n := n/2$
  else $n := 3*n+1$

Does $\pi$ terminate for arbitrary $n$? — Nobody knows!

Is there an *informative* proof of $\forall n \text{ terminate}_\pi(n) \lor \neg \forall n \text{ terminate}_\pi(n)$?

**Constructivism**

Main principle: Proofs (i.e., knowledge) should be ‘effective’:

(EE) A proof of $\exists x A(x)$ has to inform us on how to construct/find a witness for $A$.

(DE) A proof of $A \lor B$ must either show $A$ or show $B$.

Remark:

- Intuitionism is not the only school of constructive mathematics.
Constructive existence and disjunction

- EE and DE may be seen as two forms of the same principle:
  \[ \exists n A(n) \approx A(0) \lor A(1) \lor A(2) \lor \ldots \]
  \[ B \lor C \approx \exists n (n = 0 \supset B) \land (n \neq 0 \supset C) \]

- The principle also applies in contexts of (other) quantifiers:
  Proving \( \forall x \exists y A(x, y) \) effectively means to establish an algorithm \( C \),
  that for every input \( x \) outputs some \( y = C(x) \), such that \( A(x, C(x)) \)
  can be shown.
  Analogously for \( \forall x (A(x) \lor B(x)) \). [Which algorithm?]

Note:

- We speak of ‘(effective) proofs’ where classical mathematicians might just speak of ‘truth’.
- We speak of ‘algorithms’ where classical mathematicians might speak of a ‘function’ (e.g., taken as set of pairs).
- Constructive proofs always contain concrete algorithms.
  Note the obvious relevance to the systematic construction of (provably) correct software!
Brouwer’s Intuitionism

Luitzen Egbertus Jan Brouwer (1881-1966)

- Mathematics is more fundamental than logics:
  We analyze concrete mathematical arguments to understand the logic behind, not vice versa.
  Compare: logics for modelling and analyzing (programs, agents, . . .)
- Mathematical objects are (abstract, mental) constructions:
  → ‘formal methods’ is about mental constructions.

Meaning of logical symbols without reference to ‘truth’:

\(A \lor B\) means: either \(A\) or \(B\) can be shown (i.e., is known)

\(A \land B\) means: we can show \(A\) and we can show \(B\)

\(A \supset B\) means: given knowledge of \(A\), we know how to show \(B\);
  equivalently: we know how to transform a proof of \(A\) into one of \(B\)

\(\neg A =_{df} A \supset \bot\) means: we can refute \(A\); i.e., we can show that
  the assumption of knowledge of \(A\) leads to absurd claims

\(\exists x \, P(x)\) means: we can construct a \(t\) such that we can show \(P(t)\)

\(\forall x \, P(x)\) means: we know a method to show \(P(t)\) for any given \(t\)
Brouwer-Heyting-Kolmogorov (BHK) interpretation

\( \gamma \triangleright \{ A \} \ldots \gamma \) is a proof of \( A \)

\( \gamma \triangleright \{ A \land B \} \iff \gamma = \langle \gamma_1, \gamma_2 \rangle \) where \( \gamma_1 \triangleright \{ A \} \) and \( \gamma_2 \triangleright \{ B \} \)

\( \gamma \triangleright \{ A \lor B \} \iff \gamma = \langle \nu, \gamma_0 \rangle \) where

- either: \( \nu = \text{‘left’} \) and \( \gamma_0 \triangleright \{ A \} \)
- or: \( \nu = \text{‘right’} \) and \( \gamma_0 \triangleright \{ B \} \)

\( \gamma \triangleright \{ A \supset B \} \iff \gamma \) is a procedure transforming any given proof of \( A \) into a proof of \( B \)

\( \gamma \triangleright \{ \exists x \in X \ A(x) \} \iff \gamma = \langle \rho, \gamma_0 \rangle \) where \( \rho \triangleright \{ n \in X \} \) and \( \gamma_0 \triangleright \{ A(n) \} \)

\( \gamma \triangleright \{ \forall x \in X \ A(x) \} \iff \gamma \) is a procedure transforming a given proof \( \rho \) of \( n \in X \) into a proof of \( A(n) \)

\( \neg A \) is just an abbreviation for \( A \supset \bot \)

For \( \forall x \ A(x) / \exists x \ A(x) \) without range restriction: replace ‘\( \rho \)’ by ‘\( n \)’
The following formulas are intuitionistically valid:

1. $$(A \land (A \supset B)) \supset B$$
2. $$A \supset \lnot \lnot A$$
3. $$(A \land B) \supset (B \land A)$$
4. $$(A \lor B) \supset (B \lor A)$$
5. $$A \supset (B \supset A)$$
6. $$\bot \supset A$$
7. $$(A \supset (B \supset C)) \supset ((A \land B) \supset C)$$
8. $$\lnot (A \lor B) \supset (\lnot A \land \lnot B)$$
9. $$A \land (B \lor C) \supset ((A \land B) \lor (A \land C))$$
10. $$((A \land B) \lor (A \land C)) \supset (A \land (B \lor C))$$
11. $$\exists x \lnot A(x) \supset \lnot \forall x A(x)$$
12. $$\forall x A(x) \supset \lnot \exists x \lnot A(x)$$
13. $$\exists x A(x) \supset \lnot \forall x \lnot A(x)$$
14. $$\lnot \exists x A(x) \supset \forall x \lnot A(x)$$
Ad 1. \((A \land (A \supset B)) \supset B\) is valid according to BHK:
\[\gamma \triangleright \{(A \land (A \supset B)) \supset B\}\]
means: the procedure \(\gamma\) transforms every proof \(\rho\) of \(A \land (A \supset B)\) into a proof of \(B\).
\(\rho\) is a pair \(\langle \delta, \eta \rangle\), where \(\delta \triangleright \{A\}\) and \(\eta \triangleright \{A \supset B\}\).
\(\eta\) is a procedure transforming proofs of \(A\) into proofs of \(B\).
The proof (i.e., procedure) \(\gamma\) that we are looking for can now be described as follows:

1. extract the first component — i.e., the proof \(\delta\) of \(A\) — from the input \(\rho\)
2. extract the second component — i.e., procedure \(\eta\) — from \(\rho\)
3. apply \(\eta\) to input \(\delta\), i.e., compute \(\eta(\delta)\)

Note that the computed object is a proof of \(B\) \((\eta(\delta) \triangleright \{B\})\).

Therefore we have indeed presented a procedure that converts any proof of \(A \land (A \supset B)\) into a proof of \(B\).
Ad 2. \( A \supset \neg\neg A \) is valid according to BHK:

We prove more generally: \( A \supset ((A \supset B) \supset B) \)

Why is this more general than \( A \supset \neg\neg A \)? Answer yourself!

We define the procedure \( \gamma \triangleright \{ A \supset ((A \supset B) \supset B) \} \) as follows:

\( \gamma \) waits for input \( \delta \), where \( \delta \triangleright \{ A \} \), and returns as output the following procedure \( \pi \):

input for \( \pi \): \( \eta \), where \( \eta \triangleright \{ A \supset B \} \)

Note: \( \pi \) has to convert any proof of \( A \) into one for \( B \). Therefore:

output of \( \pi \): \( \eta(\delta) \), i.e., a proof of \( B \), as required

Remark: The outlined proof objects can be formalized as terms of the \( \lambda \)-calculus: \( \lambda \)-terms can be seen as functional programs.

Exercise 46:
Show that formulas 3, 4, 5, 6, 7 and 9 (on slide 8) are BHK-valid.
Elements of intuitionistic logic - invalid formulas

The following formulas are classically valid, but intuitionistically invalid:

1. \[ A \lor \neg A \]
2. \[ \neg \neg A \supset A \]
3. \[ (A \supset B) \lor (B \supset A) \]
4. \[ ((A \supset B) \supset A) \supset A \] (Peirce’s law)
5. \[ \neg (A \land B) \supset (\neg A \lor \neg B) \]
6. \[ \neg (A \lor B) \supset (\neg A \land \neg B) \]
7. \[ ((A \supset B) \land (\neg A \supset B)) \supset B \]
8. \[ \neg \forall x \neg A(x) \supset \exists x A(x) \]
9. \[ \neg \exists x \neg A(x) \supset \forall x A(x) \]
10. \[ \neg \forall x A(x) \supset \exists x \neg A(x) \]

We have seen why \( A \lor \neg A \) is not constructively/intuitionistically valid: Think of \( \forall n \text{ terminate}_\pi(n) \lor \neg \forall n \text{ terminate}_\pi(n) \).

**Exercise 47:**
For 2,3,5,7,10: argue for intuitionistic invalidity (informally).
Better ways to establish intuitionistic validity: Proof systems for intuitionistic logic

1. Hilbert-type systems for intuitionistic logic: [e.g., by Arendt Heyting, ...]
   see handout — first-order systems: + generalization: \( \frac{A(u)}{\forall x A(x)} \)

2. ‘Natural deduction’ [Gerhard Gentzen 1934/35]

3. Sequent calculus \( LI \) (aka. \( LJ \)) [Gerhard Gentzen 1934/35]

Remarks:

ad 1. Note that modus ponens and generalization are justified by the BHK interpretation.

ad 2. Gentzen attempted to model directly concrete (constructive) mathematical reasoning (proofs).

ad 3. For Gentzen a kind of meta-calculus ("Schlussweisen-Kalkül"). \( LI/LK \) allow to characterize the relation between classical and intuitionistic logic nicely.
Cut-free sequent calculi are useful for proof search!