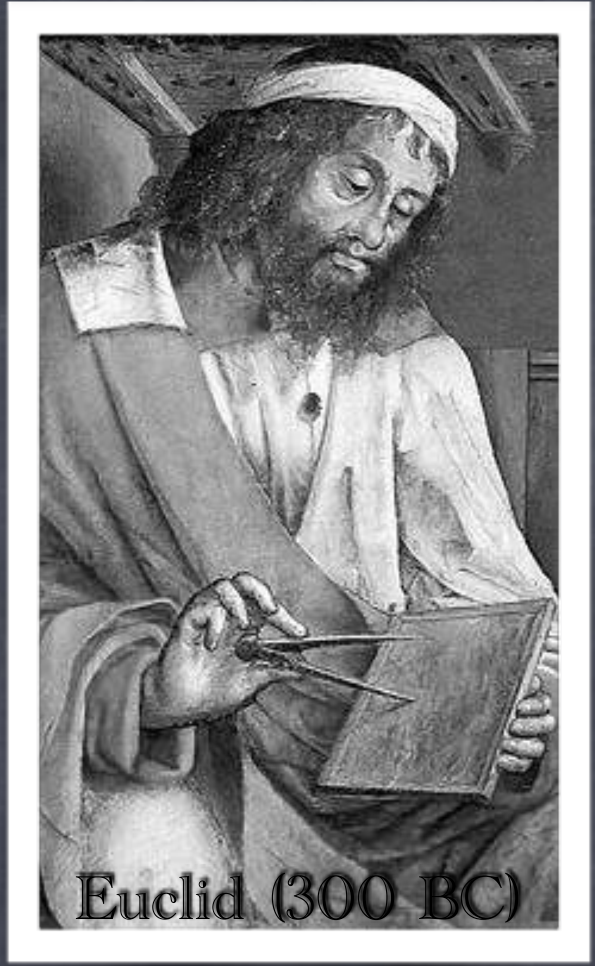


Proofs: Logic in Action

Using Logic

- Logic is used to deduce results in any (mathematically defined) system
  - Typically a human endeavour (but can be automated if the system is relatively simple)
- Proof is a means to convince others (and oneself) that a deduced result is correct
  - <u>Verifying</u> a proof is meant to be easy (automatable)
  - Coming up with a proof is typically a lot harder (not easy to fully automate, but sometimes computers can help)

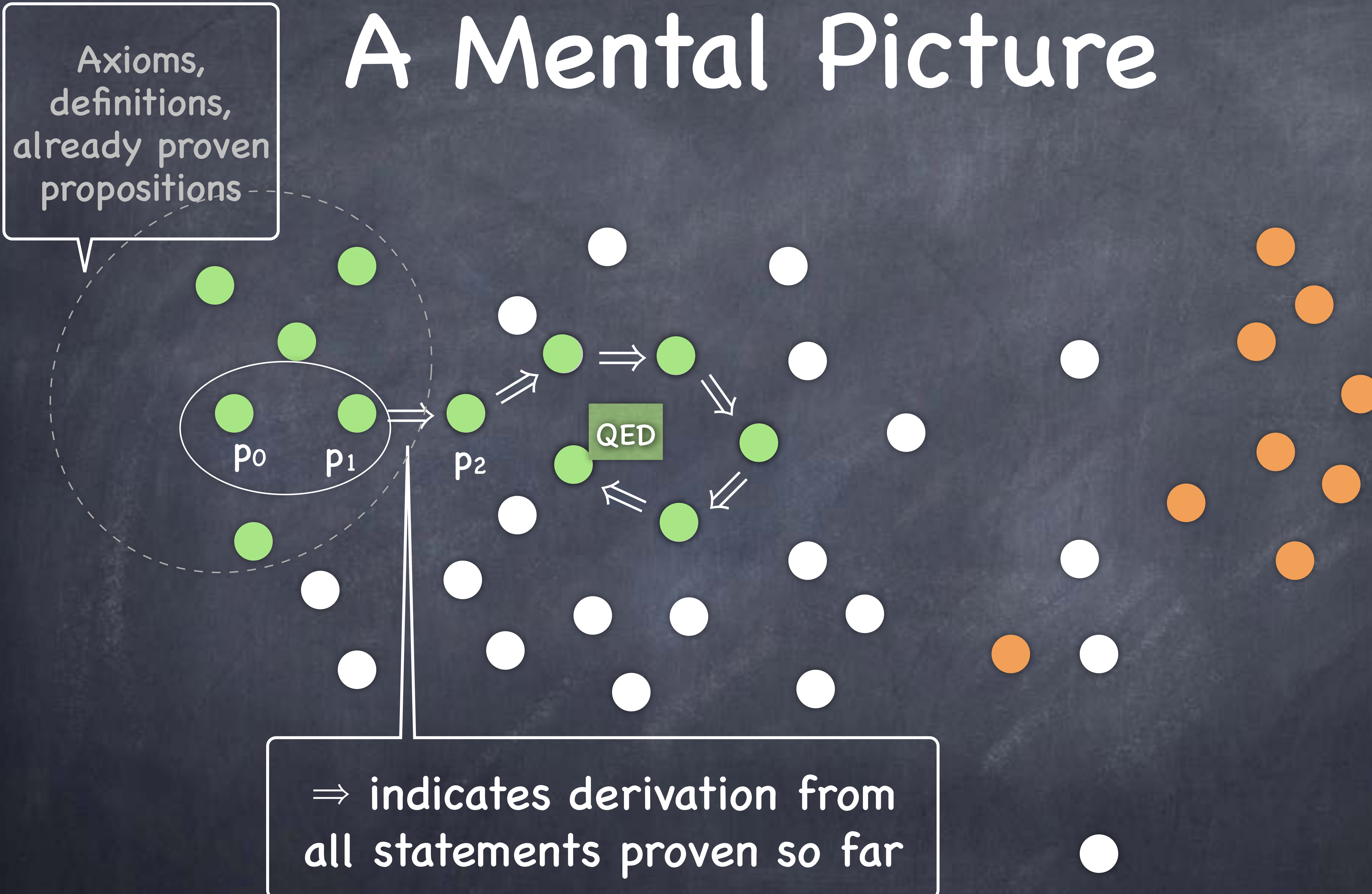
What are we proving?

- We are proving <u>propositions</u>
  - Often called Theorems, Lemmas, Claims, ...
- Propositions may employ various <u>predicates</u> already specified as <u>Definitions</u>
  - e.g. All positive even numbers are larger than 1
    - $\forall x \in \mathbb{Z}$ (<u>Positive</u>(x) $\wedge$ <u>Even</u>(x)) $\rightarrow$ <u>Greater</u>(x,1)
- These predicates are specific to the system (here arithmetic). The system will have its own "axioms" too (e.g., $\forall x\ x+0=x$)
  - For us, numbers (integers, rationals, reals) and other systems like sets, graphs, functions, ...

Anatomy of a Proof

- Clearly state the proposition p to prove (esp'ly, if rephrased)
- Derive propositions p_0, …, p_n where for each k, either p_k is an axiom or an already proven proposition in the system, or $(p_0 \land p_1 \land \ldots \land p_{k-1}) \rightarrow p_k$ holds (i.e., is True)
  - Usually one or two propositions so far would imply the next
    - [verify!] if $(p_i \land p_j) \rightarrow p_k$, then $(\ldots \land p_i \land \ldots \land p_j \ldots) \rightarrow p_k$
  - An explanation should make it easy to verify the implication (e.g., "By p_j and p_{k-1}, we obtain p_k")
- p_n should be the proposition to be proven
- May use "sub-routines" (lemmas)
  - e.g., Derive p_0, …, p_{k-1}. Let p_k be a lemma proven separately. Say, $p_k \equiv p_{k-1} \rightarrow p$. Now, let p_{k+1} be p, as $(p_{k-1} \land p_k) \rightarrow p$ holds.

A Mental Picture



Example

- Our system here is that of integers (comes with the set of integers $\mathbb{Z}$ and operations like +, -, *, /, exponentiation…)
  - We will not attempt to formally define this system!
- Definition: An integer x is said to be odd if there is an integer y s.t. x=2y+1
  - $\forall x \in \mathbb{Z}\ Odd(x) \leftrightarrow \exists y \in \mathbb{Z}\ (x=2y+1)$

"if" used by convention; actually means "iff"

- Proposition: If x is an odd integer, so is x^2
  - $\forall x \in \mathbb{Z}\ Odd(x) \rightarrow Odd(x^2)$

Example

- Def: $\forall x \in \mathbb{Z}$ Odd(x) $\leftrightarrow$ $\exists y \in \mathbb{Z}$ $(x = 2y+1)$
- Proposition: $\forall x \in \mathbb{Z}$ Odd(x) $\rightarrow$ Odd(x^2)
- Proof: (should be written in more readable English)
  - Let x be an arbitrary element of $\mathbb{Z}$. Variable x introduced.
  - Suppose Odd(x). Then, we need to show Odd(x^2).
  - By def., $\exists y \in \mathbb{Z}$ $x=2y+1$. So let $x=2a+1$ where $a \in \mathbb{Z}$. Variable a.
  - Then, $x^2 = (2a+1)^2 = 4a^2 + 4a + 1$
    $= 2(2a^2+2a) + 1$. From arithmetic.
  - $\exists w \in \mathbb{Z}$ $(2a^2+2a)=w$. From arithmetic.
  - So let $2a^2+2a=b$, where $b \in \mathbb{Z}$ Variable b.
  - Hence, $x^2 = 2b+1$
  - Then, by definition, Odd(x^2).
  - Hence for every x, Odd(x) $\rightarrow$ Odd(x^2). QED.

Proving vs. Verifying

- Proofs should be easy to verify. All the cleverness goes into finding/writing the proof, not reading/verifying it!

"**P** *vs.* **NP**" *(informally) :*
P = *class of problems for which* *finding* *a proof is computationally easy.*
NP = *class of problems for which* *verifying* *a proof is computationally easy.*
We believe that many problems in **NP** *are not in* **P**
(but we haven't been able to prove it yet!)

- Multiple approaches:
  - Direct deduction; Rewriting the proposition, e.g., as contrapositive; Proof by contradiction; Proof by giving a (counter)example, when applicable; Mathematical Induction.