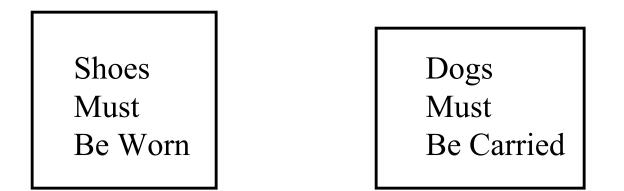
## Ambiguity



- Common signs on escalators in the UK
- What do they mean?
  - if I carry two pairs of new shoes I just bought?
  - if I have no shoes?
  - if I have no dog?

Formalisms to the Rescue?

- Use the Predicate Calculus:
- For All x (OnEscalator(x) -> Exists y such that (PairOfShoes(y) AND
  IsWearing(x,y))

and for the second sign we'll have:

For all x ((On Escalator(x) AND IsDog(x)) -> IsCarried(x))

## Do Dogs Have to Wear Shoes?

- The formalism did not automatically remove ambiguity
  - do dogs have to wear shoes?
  - what counts as a pair of shoes?
  - what counts as "wearing" shoes?
    - there are ways to fix this, but it shows formalisms are not magic
- Did you notice the signs read "must"?
  - this is known as "optative" mood (the language of requirements, normative concerns)
  - but the formalism is in the "indicative" mood

## Refutability

- To "refute" an assertion is to demonstrate that it is wrong
  - not merely a competing assertion, to really show it incorrect
- All respectable *scientific theories* are refutable
  - this supports peer review for correctness
- A theory that is not refutable will not be taken seriously
- Software Requirements must be refutable! (Science in what we do?)
  - A domain description of the system's environment or domain claims to describe the way things are
    - it should be written to invite counterexamples!
  - A requirement claims to describe how things ought to be when the system is installed
    - the customer should be able to look at it and say, "NO, that isn't what I want."
    - OR, "yes, this is the right requirement, but your product has failed to meet it."

## Designations

- A designation singles out some sort of phenomena and tells you how to recognize it and gives it a name
  - note: a "definition" can be neither true nor false
- All designations in your requirements document must be reliably and unambiguously recognizable
  - you should never be able to weasel out of a refutation by saying, "it all depends on what you mean by *payment*." (fill in your favorite observable your document)
    - if this does happen, you need to fix the designation
- You proceed to write requirements by describing relationships among the designated phenomena, the entities and data items of interest.
  - you will run the risk of refutation because your readers can use your designations to pin you down and give a counterexample.

 Your reader can say something like, "This is a motor vehicle, right? And here are the <u>roadwheels</u> as you've defined them. Your description says all <u>motor vehicles</u> have an even number of <u>roadwheels</u>. But here, your user has a <u>motor vehicle</u> with 3 <u>roadwheels</u>. So you're wrong.