

On being the same as.

why something so simple is so hard.

Pat Hayes, Florida IHMC

Invited talk, UDC Seminar on
Classification and Ontology
September 2011, The Hague

Logic

- The semantic web formalisms are all slimmed-down versions of modern logic
- Which therefore share some of modern logic's basic assumptions about the world it describes.
- Logic's world is made up of *individual particulars* and sets of them.

Individual particulars

- “Individual” is not a classifier. *Anything* can be (treated as) an individual.
- But individuals are required to be **unique** and **distinct**. There can be no doubt of the matter as to how many of them are in a set, for example.
- Particulars don’t have instances. (Well, maybe they do, but that doesn’t play any role in the logic.)

Logic's view of the world

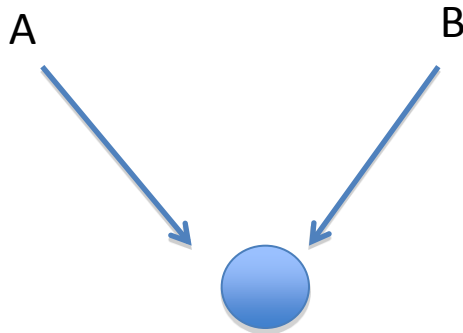
- Logical names *refer to* particular individuals.
- It is the referent of a name which determines the truth of sentences containing that name: truth depends on reference (and nothing else).
- Any other role or function of names is stripped away from the logical picture.
- Properties and relations are basically *sets* of particulars.

Logical equality

- A sameAs B is true when the referent of 'A' is the same as the referent of 'B'

Logical equality

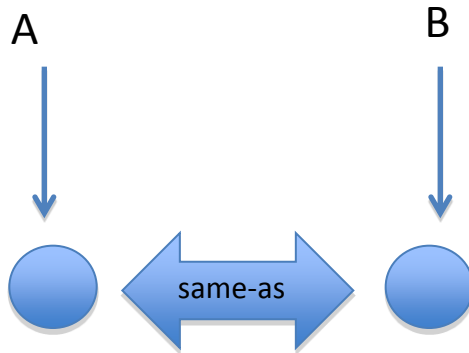
- $A \text{ sameAs } B$ is true when the referent of 'A' is the same as the referent of 'B'



There is ONE thing that both names refer to.

~~Logical equality~~

- $A \text{ sameAs } B$ is true when the referent of 'A' is in the *same-as* relationship to the referent of 'B'



There are TWO things with a 'sameness relationship' between them. NOT equality.

People often get this wrong...

- E.g. the owl.owl ontology says

```
<rdf:Property rdf:about="#qualifierCardinality" />  
<rdfs:range rdf:resource="&xsd;nonNegativeInteger" />  
</rdf:Property>  
  
<rdf:Property rdf:about="#owl:sameAs">  
<rdfs:comment>The property that determines that two given individuals are equal.</rdfs:comment>  
<rdfs:domain rdf:resource="#owl:Thing" />  
<rdfs:isDefinedBy rdf:resource="#owl:" />  
<rdfs:label>sameAs</rdfs:label>  
<rdfs:range rdf:resource="#owl:Thing" />  
</rdf:Property>  
  
<rdf:Property rdf:about="#owl:someValuesFrom">  
<rdfs:comment>The property that determines the class that an existential property restriction refe  
<rdfs:domain rdf:resource="#owl:Restriction" />
```

Q: When are two individuals equal? A: Never.

Substitutivity is inevitable.

- **Given:** It is the referent of a name which determines the *truth* of sentences containing that name: truth depends on reference (and nothing else).
- **And:** A sameAs B is true when the referent of 'A' is the same as the referent of 'B'
- Then it follows **necessarily** that:

A sameAs B entails ([... A ...] \Leftrightarrow [... B ...])

Which is to say, if there is **any** sentence [... ? ...] with {...A...} but not {...B...}, then A is *not* sameAs B.

So then there must be at least two of them.

Does this matter?

- Logic is very flexible indeed about what counts as an 'individual'. Ghosts, sensory data, abstract things, nonexistent things, entire worlds, sets, relations,... can all be individuals. But when they are, they are all treated equally, as particulars, with the crispest possible criteria of identity.
- The world often permits a variety of ways to be individualized. In order to accommodate this in a logic, we have to embrace them all. So the conceptual world fills up with many different entities all corresponding to the same real thing: Paris the city (now), Paris the history of a city, Paris the administrative capital, Paris the artistic center, Paris last week, ... These must be distinguished in the logic, yet they are clearly in some sense all the 'same thing'. (In more than just being named by the same name.)

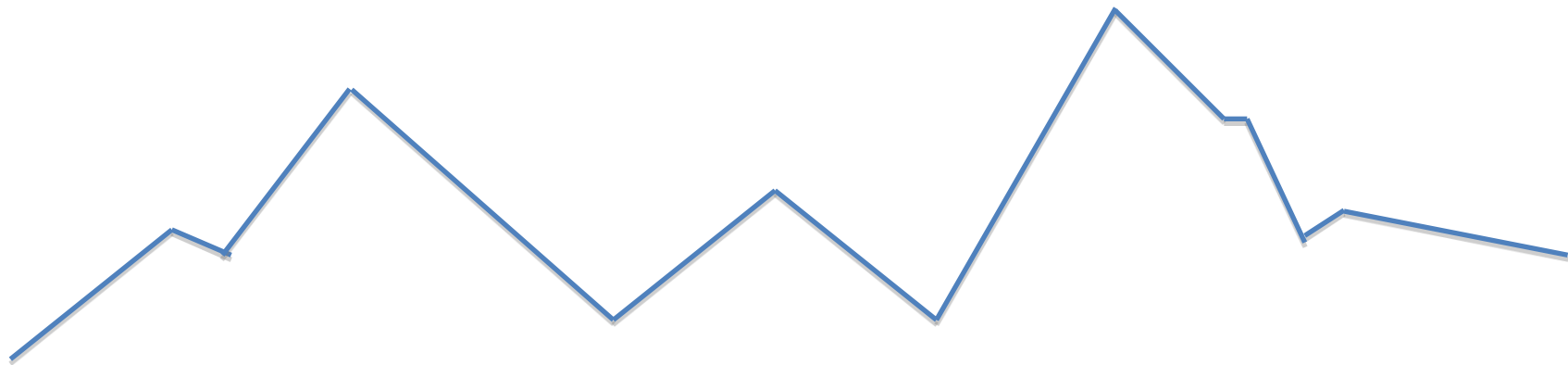
Logic vs. Occam

- Suppose we distinguish between *continuants* and *occurrents*. (Many people do, eg OBO)
- A person can be looked at either way, so we have *persons-considered-as-occurrents* and *persons-considered-as-continuants*, and axioms describing the relationships. These are never equal to one another.
- So when Joe goes into an empty room, there are *two* things in the room. (?!)
- The logic has imposed an ontological multitude on a simple world. What we need, but do not have, is a way to say: these different “things” are not really different things in the *actual* world, but are different ways that our logic views one actual thing.

Logic vs. Occam

- Dean Crawford, PhD. is a powerful figure in the University.
- Ms. Crawford is Julie's mom who picks her up late on Tuesdays.
- Juliet 'Babsicle' is Mr. Crawford's wife.
- There are many settings where it would not be *correct* to use one name rather than the other.
- We could take this outside the logic altogether (correct \neq true), but if we don't, then we need at least three Juliet Crawfords.

Counting 'things' in a fractal world



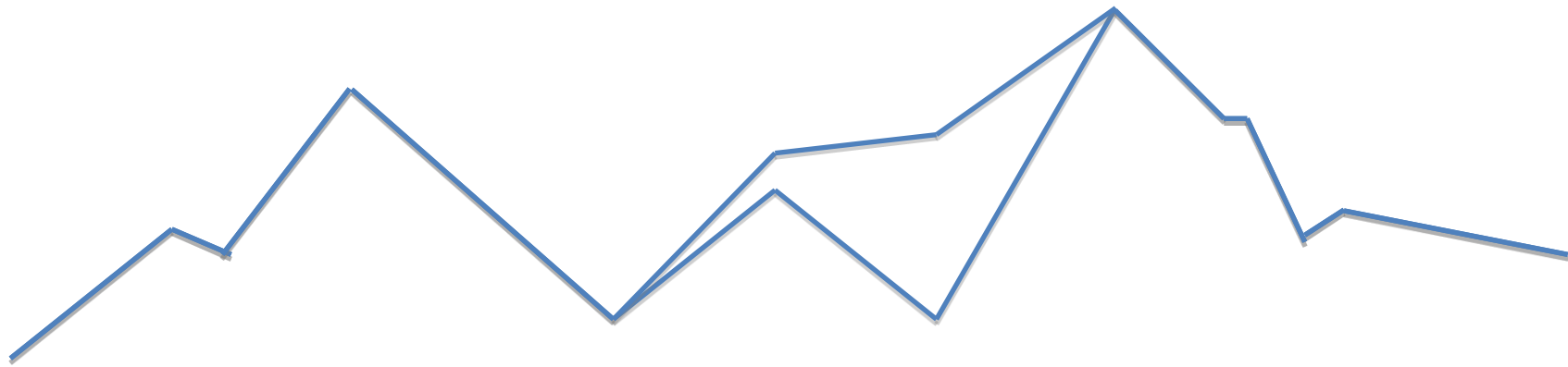
UDC Seminar "Classification and Ontology",
The Hague, September 2011. Pat Hayes,
Invited talk.

Counting 'things'



UDC Seminar "Classification and Ontology",
The Hague, September 2011. Pat Hayes,
Invited talk.

Counting 'things'



UDC Seminar "Classification and Ontology",
The Hague, September 2011. Pat Hayes,
Invited talk.

Fractals ruin individuation

- How long is the coastline of Britain?
- How many lakes are there in Norway?
- These questions have no “real” answer. There is no fact of the matter: it depends upon how you view it.

Frege's puzzle

- I tell you that the morning star is the same thing as the evening star.
- If true, then these names must co-refer.
- But you did gain information from learning this, so you must be understanding the names not purely referentially.
- Frege: “morning star” and “evening star” have different *senses*.

What is the *sense* of a name?

- When we hear a name and 'understand' it, what exactly are we understanding? Apparently not (just) the referent.
- Sometimes we can figure out the referent from the name itself, eg numerals.
- But not often.

A conversation.

- I met Joan today at the dentist.
- = Joan?
- Yes, you know, *Joan*. We met her at the Smith's party last week.
- = The tall dark woman with the extraordinary dress?
- No, the redhead. Talkative.
- = Oh, *that* one. She kept on and on about her cousin's dog catching fire?
- Yes, her. Joan.
- = I never did catch her name. What about her?
- I met her again today, at the dentist.
- = Ah.

What happened in line 6?

Locating in a narrative

- The function of the name use here is to *locate a character in narrative memory*. This is the 'aha' moment of grokking the sense.

- The function of using a name in communication is to *identify a point in a the listener's knowledge space sufficiently precisely to provide a hook for subsequent information about the referent.*
- This can be done without identifying the referent, and it can fail even when the referent is known.
- It really has almost nothing to do with reference in the logical sense.

Locating in general knowledge

- So, we were driving through Languedoc...
= Languedoc?
- It's a region in France.
= Uh-huh.
- ...and of course we couldn't help noticing the chimneys...
= Chimneys?

What's in a name?

Bill and Sue are married. Bill plays lacrosse. Robert knows them both but thinks they are not married. Mary does not know anything at all about Bill. She understands the name "Bill" to refer to Robert, and she knows that Robert and Sue are not married. Mary's sister Joan knows Bill personally, but also, for reasons that need not detain us here, believes that his name is "Robert". Finally, Joan's friend Wilma, a lacrosse fan, knows Bill as a lacrosse player and knows that Robert is a friend of Joan but, unlike Joan, she does not know that these are in fact the same person.

What's in a name?

Bill and Sue are married. Bill plays lacrosse. Robert knows them both but thinks they are not married. Mary does not know anything at all about Bill. She understands the name "Bill" to refer to Robert, and she knows that Robert and Sue are not married. Mary's sister Joan knows Bill personally, but also, for reasons that need not detain us here, believes that his name is "Robert". Finally, Joan's friend Wilma, a lacrosse fan, knows Bill as a lacrosse player and knows that Robert is a friend of Joan but, unlike Joan, she does not know that these are in fact the same person.

(married Bill Sue)

(believes Robert (that (not (married Bill Sue))))

(= Robert ('Bill' (BeliefsOf Mary)))

(forall ((s charseq) (p person))(= (s (Beliefs p))(s p)))

(believes Mary (that (not (married ('Bill' Mary) Sue))))

(= Bill ('Robert' Joan))

(= Bill ('Bill' Wilma))

(not (= Bill ('Robert' Wilma)))

Context names

- ('name' context)
- $\text{name}_{\text{context}}$
- Treats a name as a function on 'contexts' whose value is a logical individual. The "real" thing is now a kind of bundle of these values. The contexts are not in the actual world, but are logical facets through which the world can be viewed.
- `sameRealThingAs` might be glossed as 'in the same bundle as'. Paris is a bundle of ways to look at the "actual" city. In a sense it is the same as all of them, but not strictly the logical `sameAs` any of them.
- But Occam says: You had N and you wanted 1. Now you have N+1. Sigh.