CEID

MSc on DATA DRIVEN COMPUTING AND DECISION MAKING (DDCDM)

KNOWLEDGE REPRESENTATION AND REASONING WITH FRAMES

I. HATZILYGEROUDIS, PROFESSOR EMERITUS

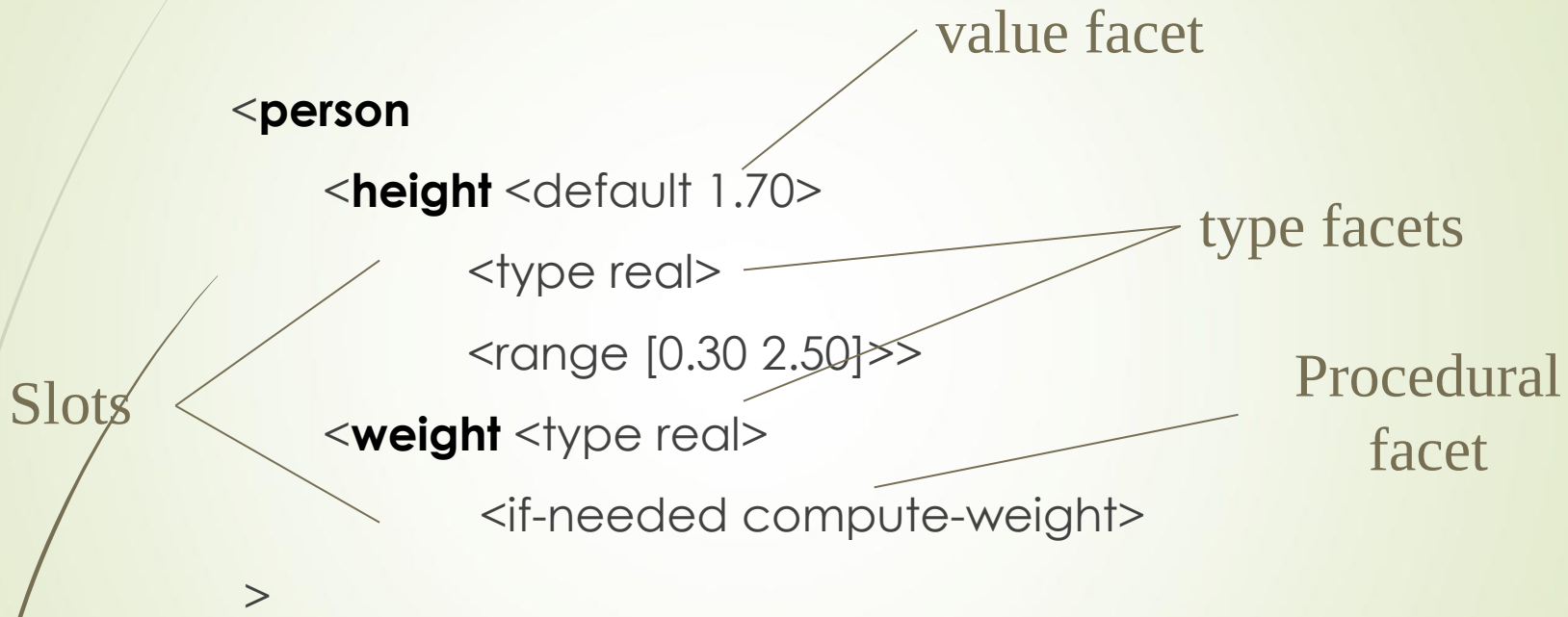
FRAMES

- They were defined by Minsky as "data structures for representing stereotyped situations". They are also called *schemata*.
- In a sense they are an evolution of semantic networks (or association networks)
- Although they require skillful and painstaking work, they have evolved into an important way of representing knowledge.

FRAMES

- Frames have:
  - **name**
  - A series of **slots** that describe properties via
  - **facets** which have **fillers** (values) and are distinguished in
    - ✓ declarative
      - o **type** (type, range)
      - o **value** (value, default)
    - ✓ procedural → procedural attachment
      - o (optional) **attached procedures**, also called *demons* (e.g., if-needed, if-added, if-removed)
      - o are activated when there are changes in a frame

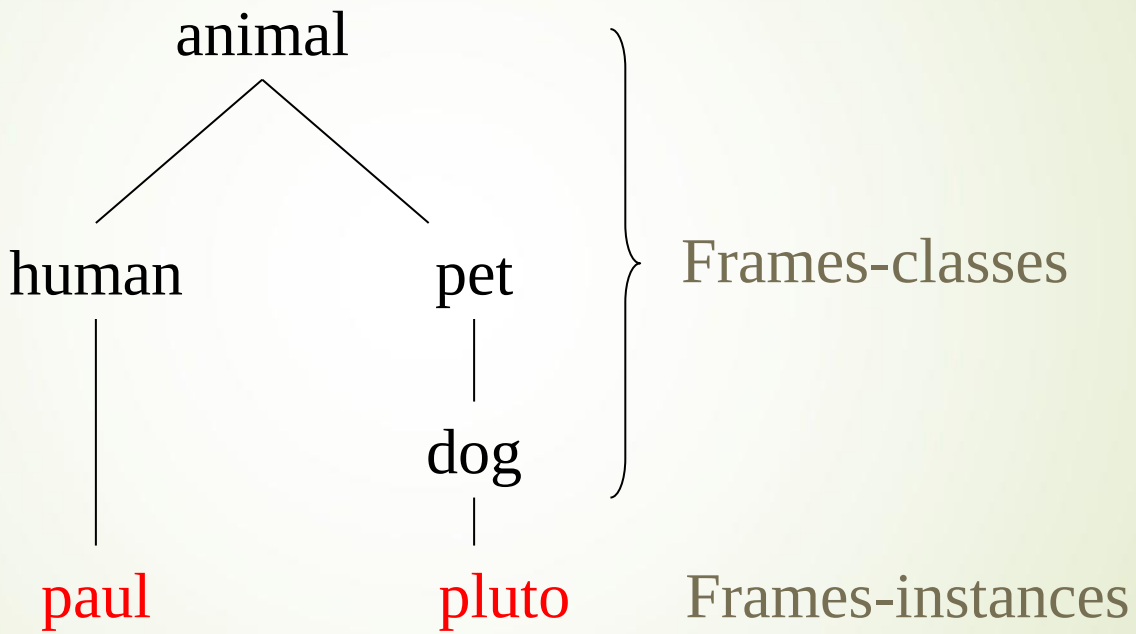
FRAMES-EXAMPLE



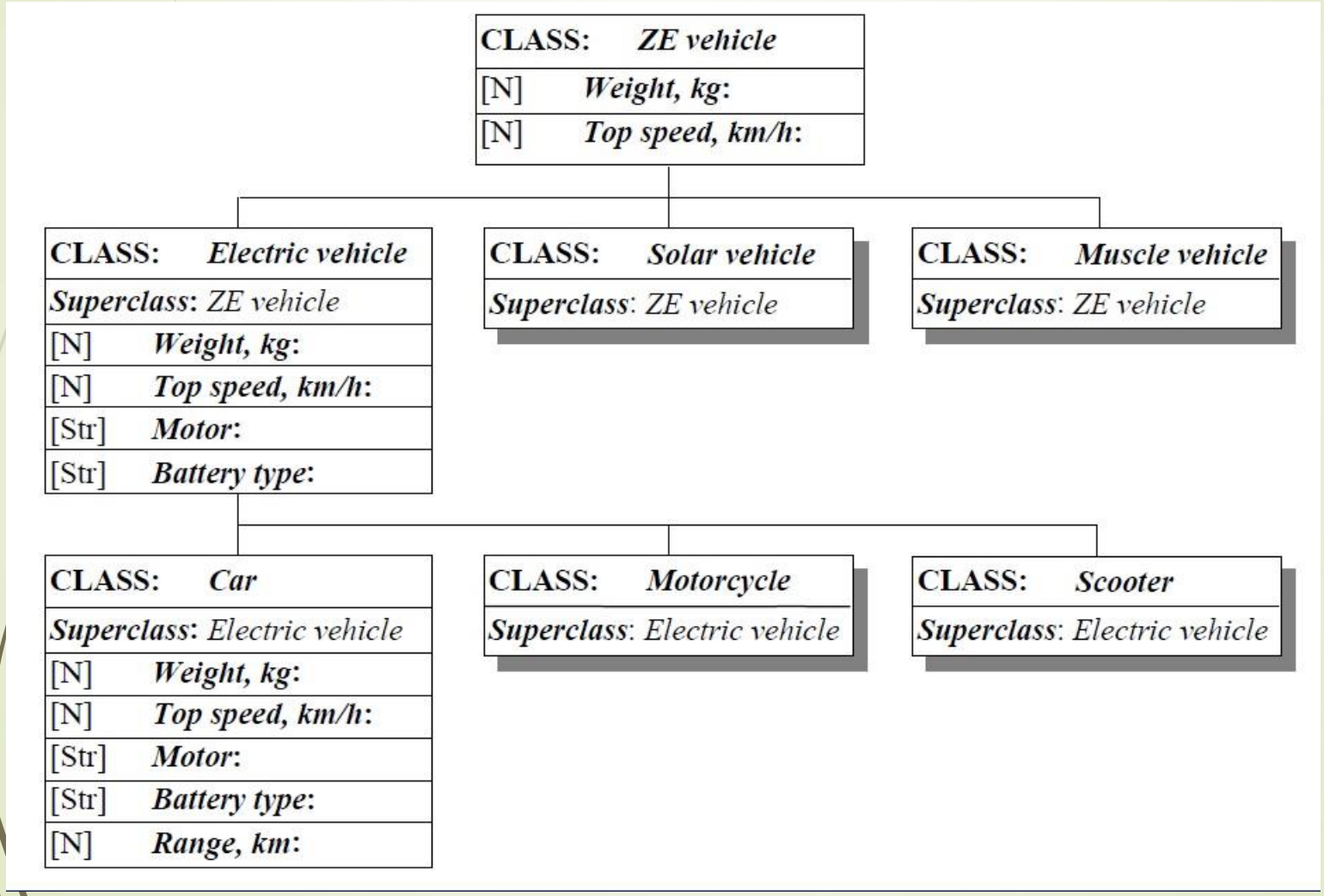
HIERARCHIES OF FRAMES

- Frames are always organized in hierarchies.
- We distinguish **frames-classes** (descriptions of general entities) and **frames-instances** (descriptions of specific entities).
- We distinguish **subframes** and **superframes** among class-frames.
- Each frame-class (except the top one) is a subframe of a more general frame-class (generalization).
- A subframe may have additional slots than its superframe (specialization).
- Each frame-instance belongs to a frame-class
- A frame-instance can have additional slots in relation to the frame-class to which it belongs.

FRAMES-EXAMPLE



FRAMES-EXAMPLE



FRAMES-FEATURES

- Frames provide:
  - ✓ inheritance,
  - ✓ procedural attachment and
  - ✓ default values
- A frame includes all the information for the specific concept it represents.
- Avoid logical inadequacy
  - ✓ The do not manifest the phenomenon of combinatorial explosion in searching for an answer.

INHERITANCE-REASONING (1)

- Each frame inherits attributes (slots) from its superframes.
- Inheritance is dynamic
- It is the only general mechanism used for inference
- Reasoning with frames means searching for the value of a property/attribute of a frame.
- The process is usually activated by starting a read or find function.

INHERITANCE-REASONING (2)

- Types of inheritance
- **Simple**: each frame-class inherits from only one superframe
- **Multiple**: each frame-class can inherit from more than one superframe
- The type of inheritance depends on the required frame hierarchy structure.

INHERITANCE-REASONING (3)

Given: Frame F, Attribute S, **Asked:** value of S

Algorithm of type N (simple inheritance)

1. Look for the value of S in the "value" facet of the S slot in the F frame.
2. If you find it, stop (success).
3. If it doesn't exist, then follow the hierarchy up (to the top) by searching each superframe for the "value" facet of slot S. If a value is found, stop (success).
4. If you don't find it, then repeat steps 1-3, this time looking at the "if-needed" facet.
5. If step 4 fails, then repeat steps 1-3, this time looking at the 'default' facet.
6. If no value found, stop (failure).

INHERITANCE-REASONING (4)

Given: Frame F, Attribute S, **Asked:** value of S

Algorithm of type Z (simple inheritance)

1. Look up the value of S in the "value", "if-needed", and "default" facets (in that order) of the S slot in the frame F.
2. If you find it, stop (success).
3. If it doesn't exist, then follow the hierarchy up (to the top) by searching each superframe for the "value", "if-needed" and "default" facets (in that order) of the S-slot. If you find a value, stop (success).
4. If you don't find it, stop (failure).

INHERITANCE-REASONING (5)

Algorithm of type N

- Gives priority to the "value" facet over the "default" facet, even if the value is higher (ie further away) in the hierarchy.

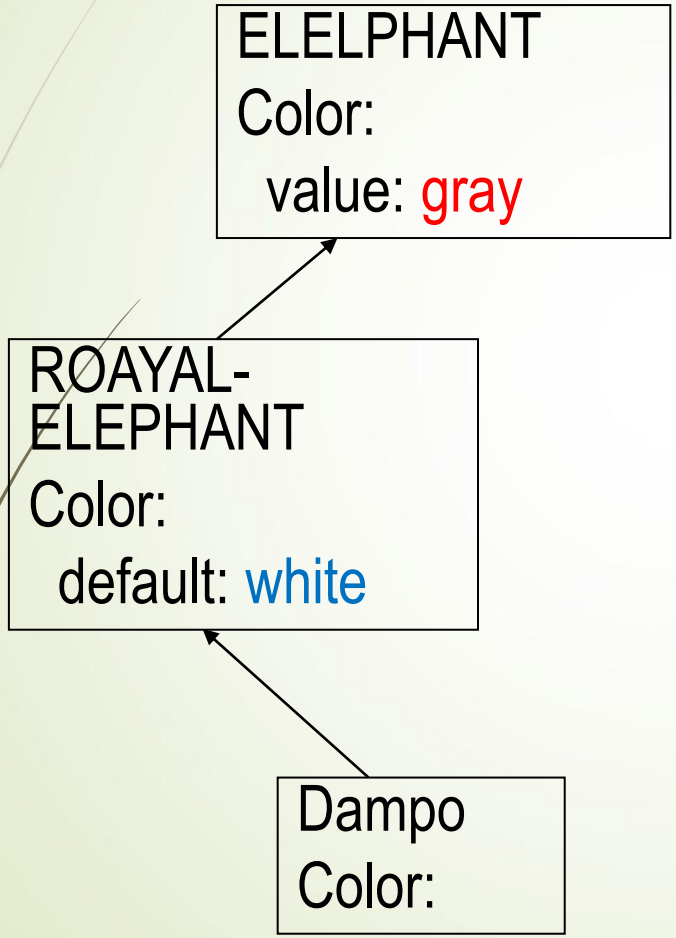
Algorithm of type Z

- Gives priority to the closest value facet, no matter whether it is of type value or default.

Combination of N and Z

- Simultaneous examination of facets «value» and «if-needed» and afterwards of «default».

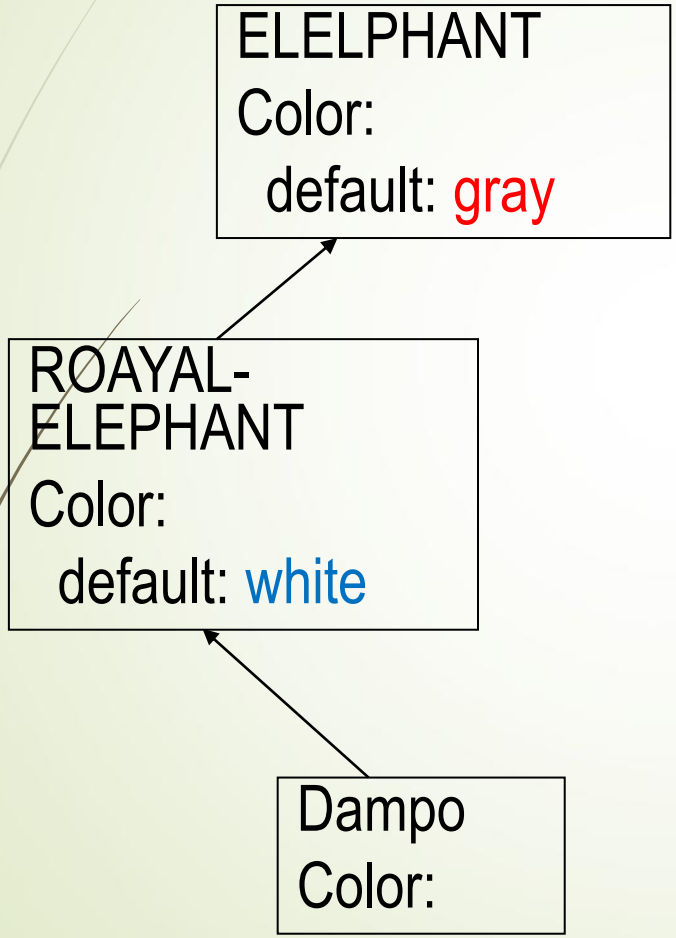
INHERITANCE-REASONING (6)



Algorithm N: gray

Algorithm Z: white

INHERITANCE-REASONING (7)

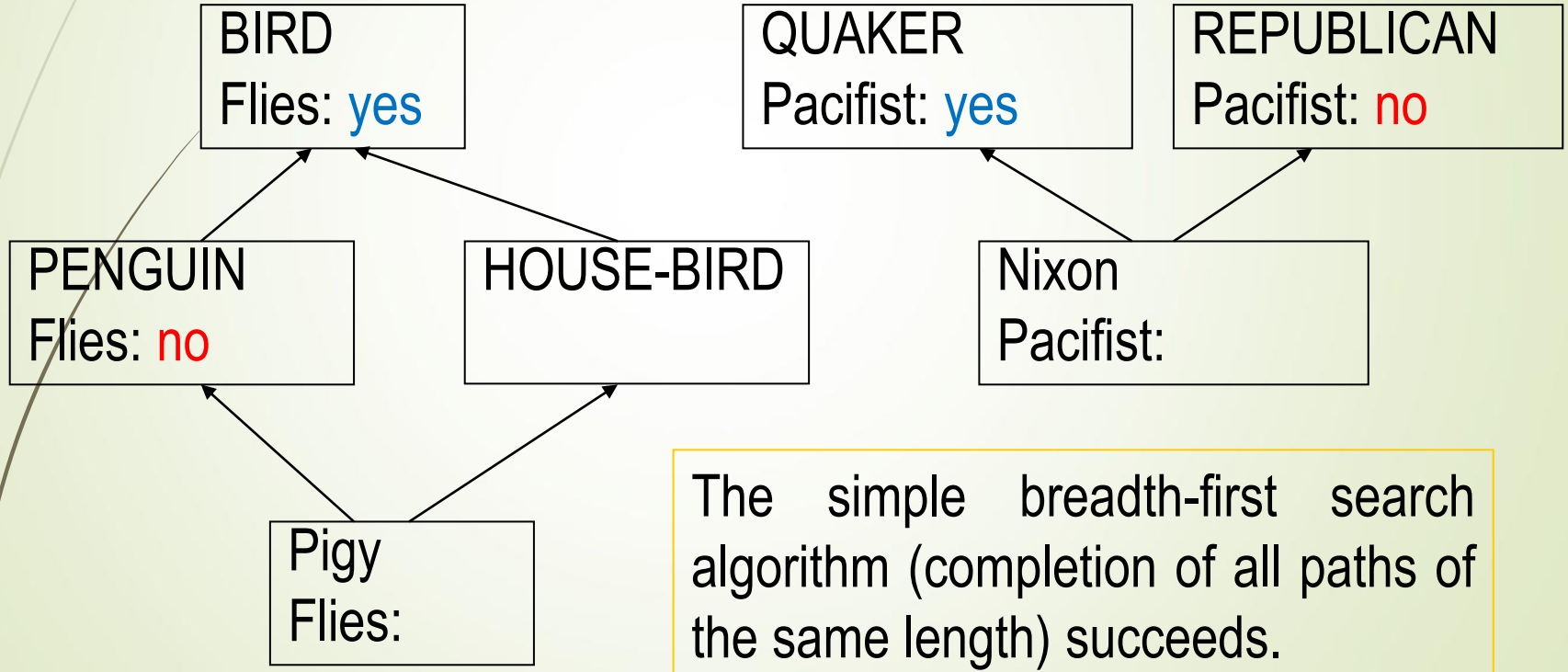


Algorithm N: white

Algorithm Z: white

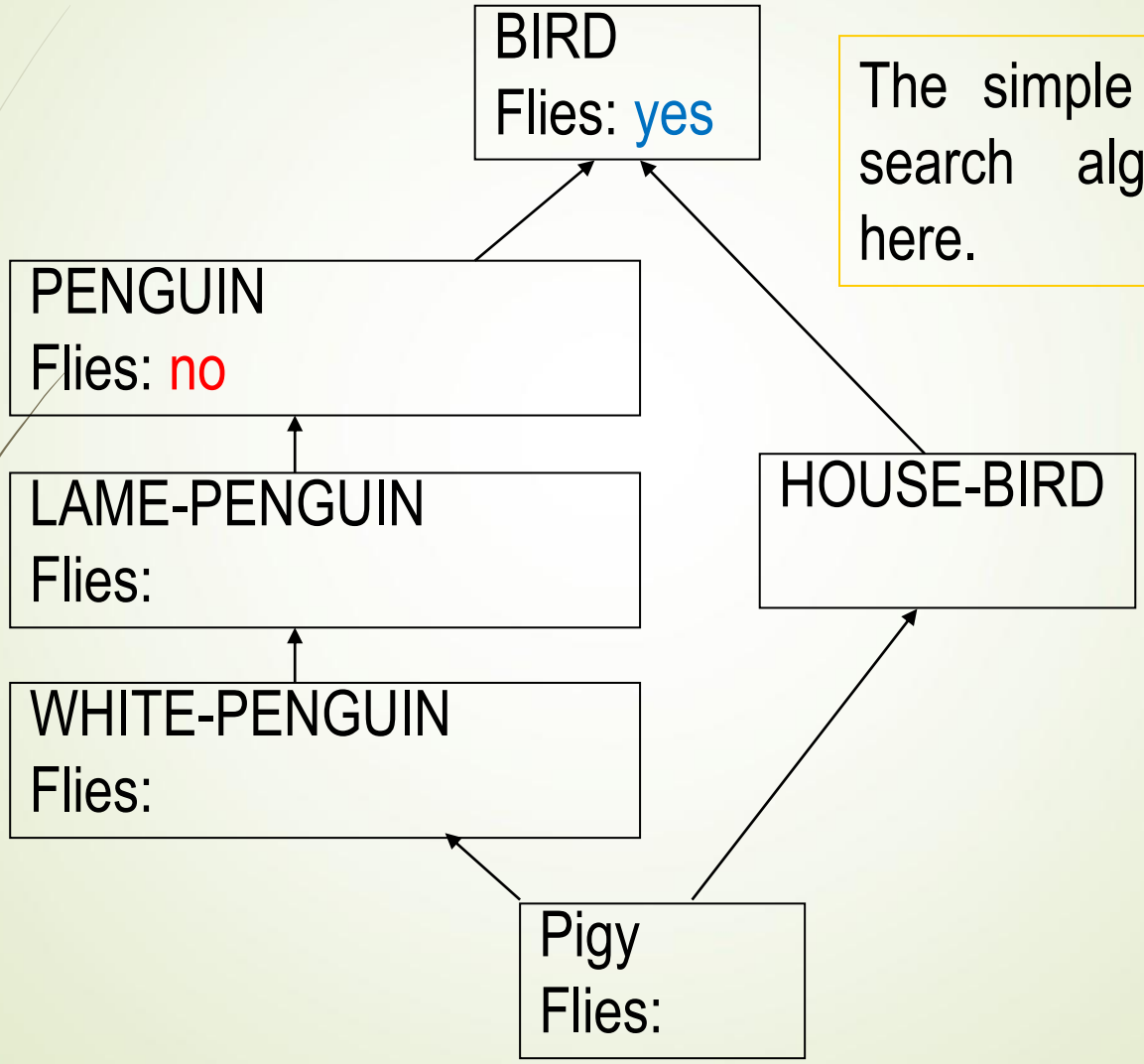
INHERITANCE-REASONING (8)

Multiple Inheritance



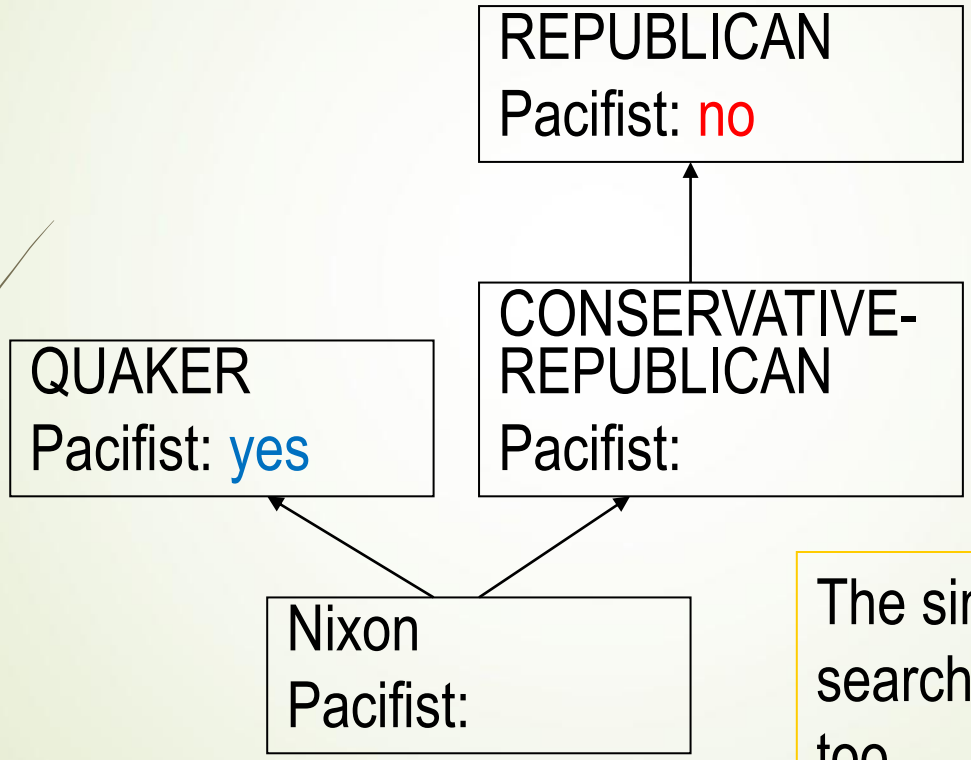
The simple breadth-first search algorithm (completion of all paths of the same length) succeeds.

INHERITANCE-REASONING (9)



The simple breadth-first search algorithm fails here.

INHERITANCE-REASONING (10)



The simple breadth-first search algorithm fails here too.

INHERITANCE-REASONING (11)

Reasoning Distance

The distance of a frame F1 from a frame F2 is less than its distance from frame F3 if and only if there is a reasoning path from F1 to F3 via F2.

.

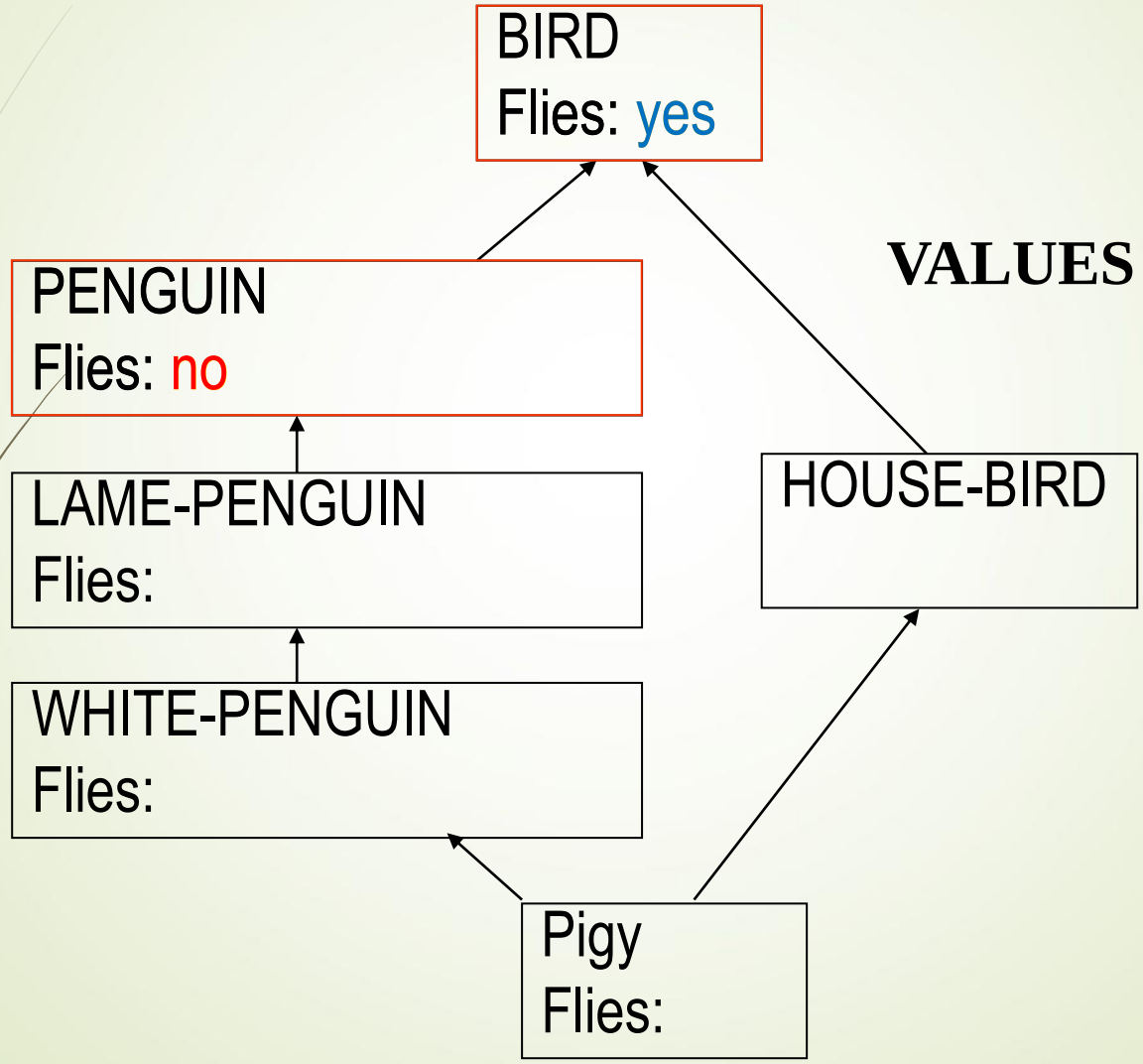
INHERITANCE-REASONING (12)

Given: Frame F, Attribute S, **Asked:** value of S

Algorithm Based on Reasoning Distance

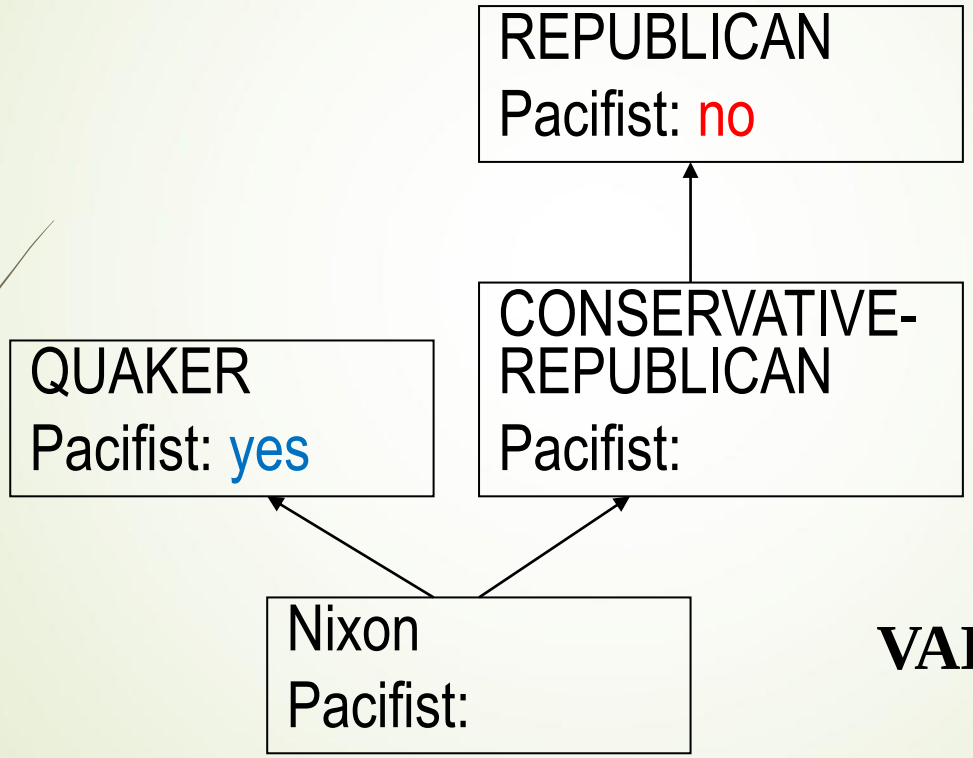
1. Apply a breadth (or depth) search following all possible paths from F upwards and store in the VALUES list all the values found for S.
2. For each value in VALUES, check if there is another value coming from a frame that has less reasoning distance from F. If there is, delete the value.
3. If there are 0 values left, then there is no answer. If one (1) value remains it is the answer. If more than one remains, then there is a contradiction.

INHERITANCE-REASONING (13)



VALUES = [no, ~~yes~~]

INHERITANCE-REASONING (14)



VALUES = [no, yes]

RULES FOR HIERARCHY DESIGN

- We describe each piece of knowledge (description of a property or attribute) once, so that we do not have multiple representations. This is achieved by putting it as higher up in the hierarchy as possible, so that it is inherited by as many subframes (subclasses) as possible. Of course, "as high as possible" is not meant in an arbitrary way, but in a way that the knowledge part is inherited by the correct subframes.

RULES FOR HIERARCHY DESIGN

- As we move down, the frames contain only the knowledge that differentiates them from their superframes. This means either descriptions of new properties or new values of already described properties. We do not repeat knowledge that is listed above and will be inherited.
- When the word "usually" is present in the value assignment to an attribute, then it is typically captured by a "default" value.

RULES FOR HIERARCHY DESIGN

- The placements of the various values should be done in such a way so that correct answers are extracted based on some algorithm. We pay particular attention to exceptions (related to default values).

RULES FOR HIERARCHY DESIGN

- Each property/attribute is described as follows:
  - ✓ <name>
  - ✓ values: <type or enumeration of values>
  - ✓ value: <real value>
  - ✓ default: <most usual value>
- Of those, <name> and the facet 'values' are required in the initial description of an attribute. The facet 'values' is not repeated further down the hierarchy.

RULES FOR HIERARCHY DESIGN

- Further down the hierarchy are descriptions that have either <name> and the 'value' facet or <name> and the 'default' facet. It makes no sense for the 'value' and 'default' facets to exist together in the same description in the same frame.

ADVANTAGES-DISADVANTAGE OF FRAMES

- ADVANTAGES
  - ✓Naturalness of representation
  - ✓High performance
  - ✓Default (or reasonable) reasoning
- DISADVANTAGES
  - ✓No clear semantics
  - ✓Limited representation