CS 295A/395D: Artificial Intelligence

Knowledge Representation

Prof. Emma Tosch

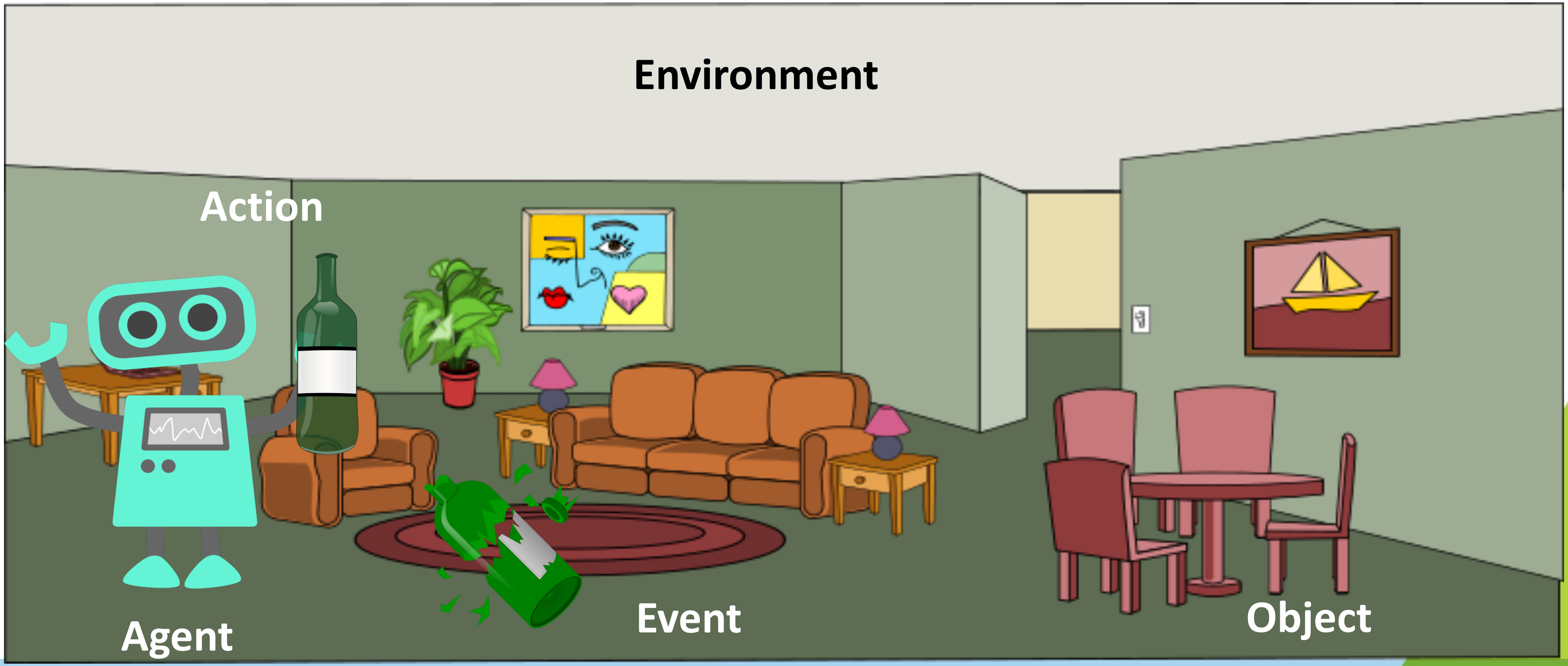
21 January 2021

The University of Vermont

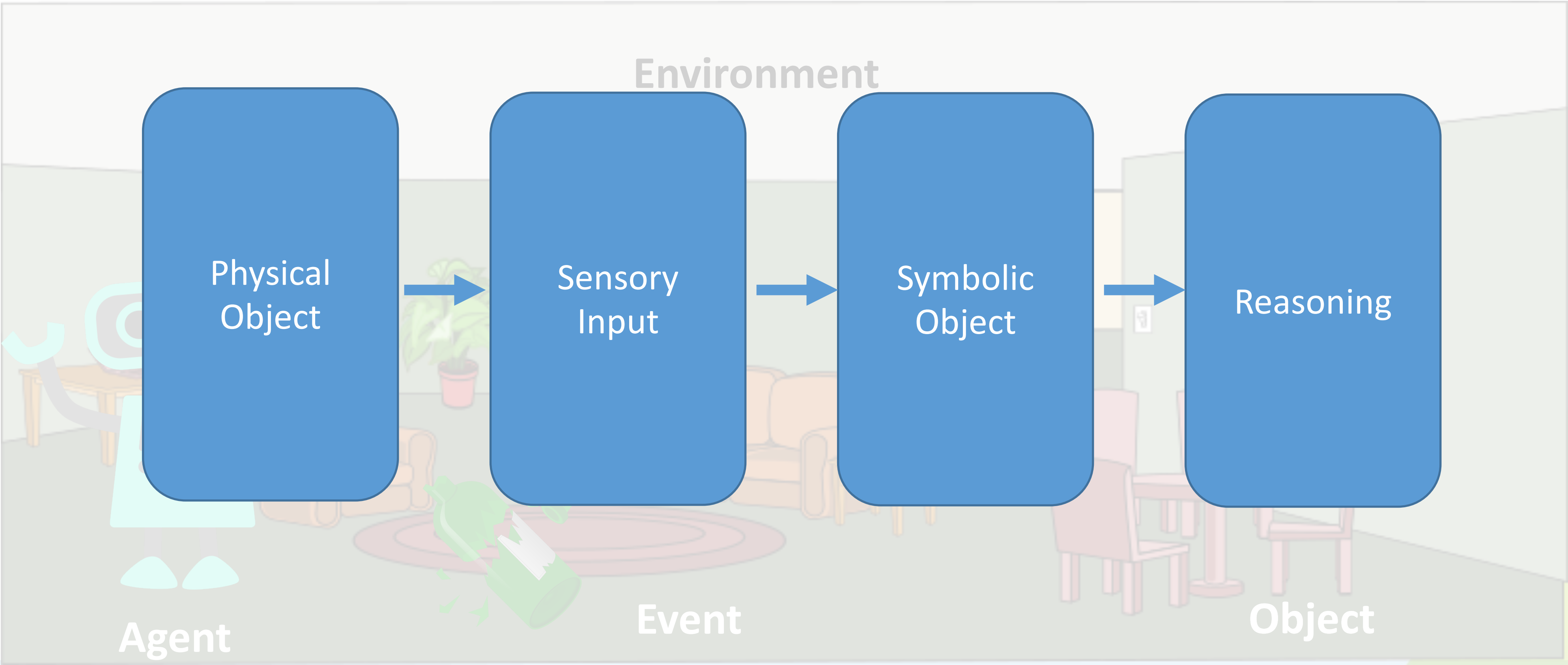
Agenda

- **Knowledge Representation: Motivation**
- Ontologies: Categories and Relations
- Background Review: Set Theory Notation
- Clicker Questions
- *Bonus*: OOP as objects in an ontology

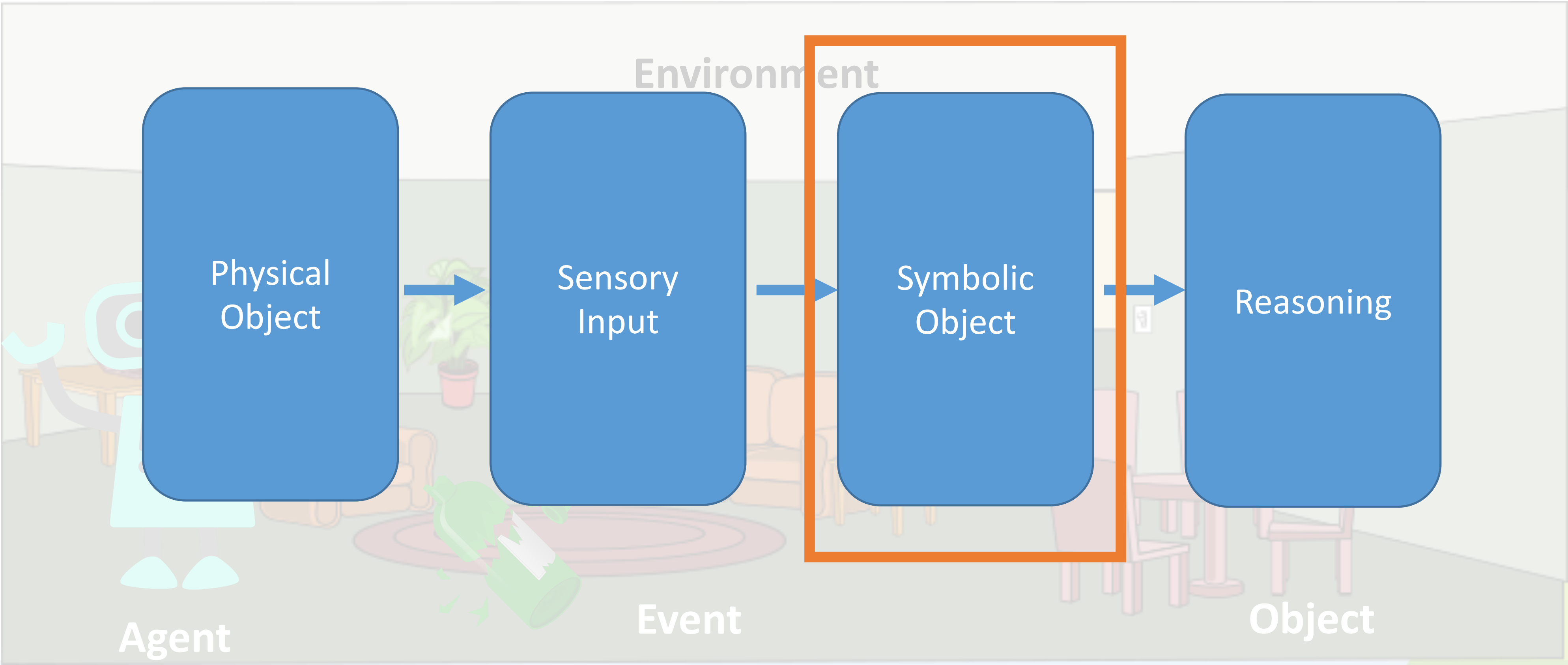
Knowledge Representation: Motivation



Knowledge Representation: Motivation



Knowledge Representation: Motivation



Agenda

- Knowledge Representation: Motivation
- **Ontologies: Categories and Relations**
- Background: Set Theory Notation
- Clicker Questions
- *Bonus*: OOP as objects in an ontology

Ontologies: Categories and relations

A ***model*** is a simplified representation of the behavior a system.

An ***ontology*** is a collection of ***categories*** and ***relations***.

A category is a collection of ***things/stuff*** that share ***relations.***

A ***relation*** is a pairing of instances from possibly different categories.

A model may use an ontology in its representation of a system's behavior.

Agenda

- Knowledge Representation: Motivation
- Ontologies: Categories and Relations
- **Background Review: Set Theory Notation**
- Clicker Questions
- *Bonus*: OOP as objects in an ontology

Set Theory: Set definition and size

A set S is a container for a collection of objects called elements.

| S | denotes the size of the set

Set Theory: Contains

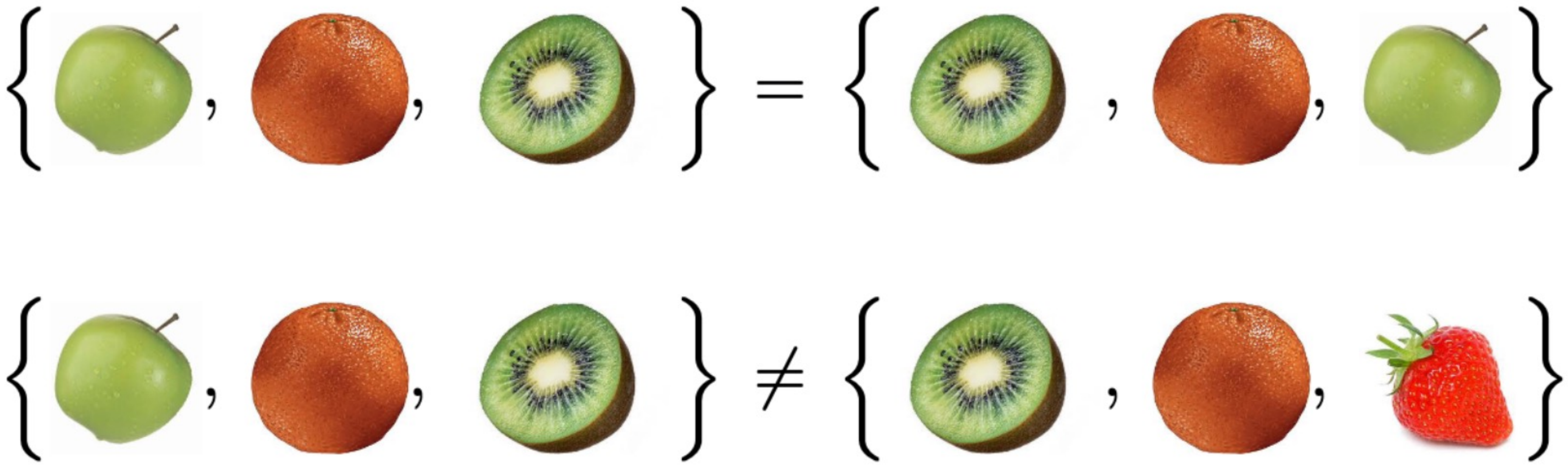
$s \in S$ is an assertion saying that s is an element of S. (e.g., In(s, S) or ElementOf(s, S))

Background: Classes of sets

- **Empty set**: The unique set that does not contain anything $S = \{ \} = \emptyset$
- **Finite sets**: Contain a finite number of elements (e.g., fruit example)
- **Countably infinite sets**: Elements can be indexed by non-negative integers (e.g., even numbers, humans into the future)
- **Uncountably infinite sets**: Sets that cannot be indexed by the non-negative integers (e.g., real numbers between 0 and 1)

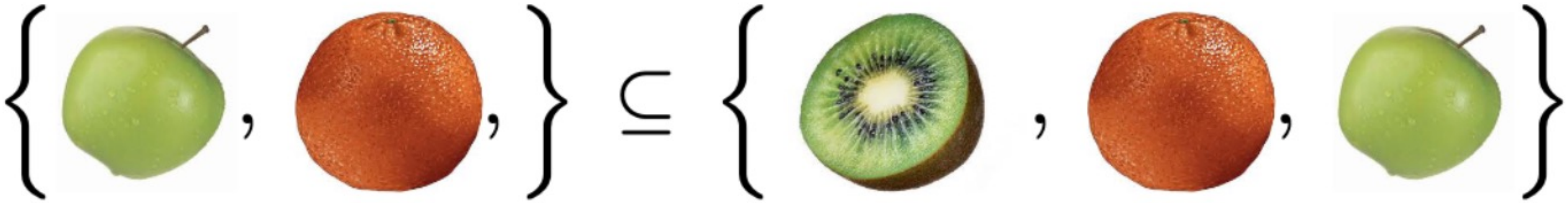
Set Theory: Equality

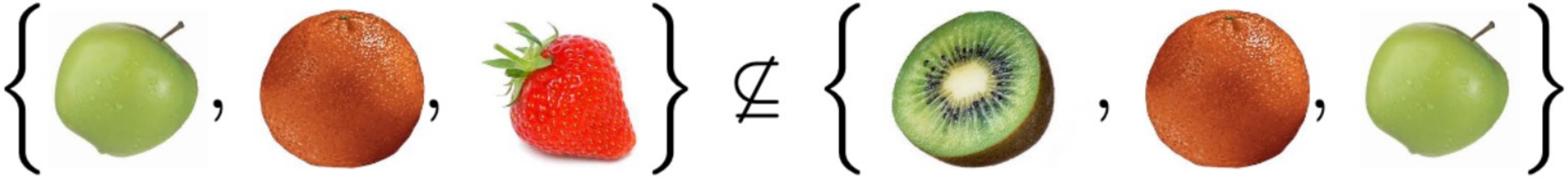
$S = T$ is an assertion that S is equal to T. Two sets are equal iff every element of S is a member of T.



Set Theory: Subset

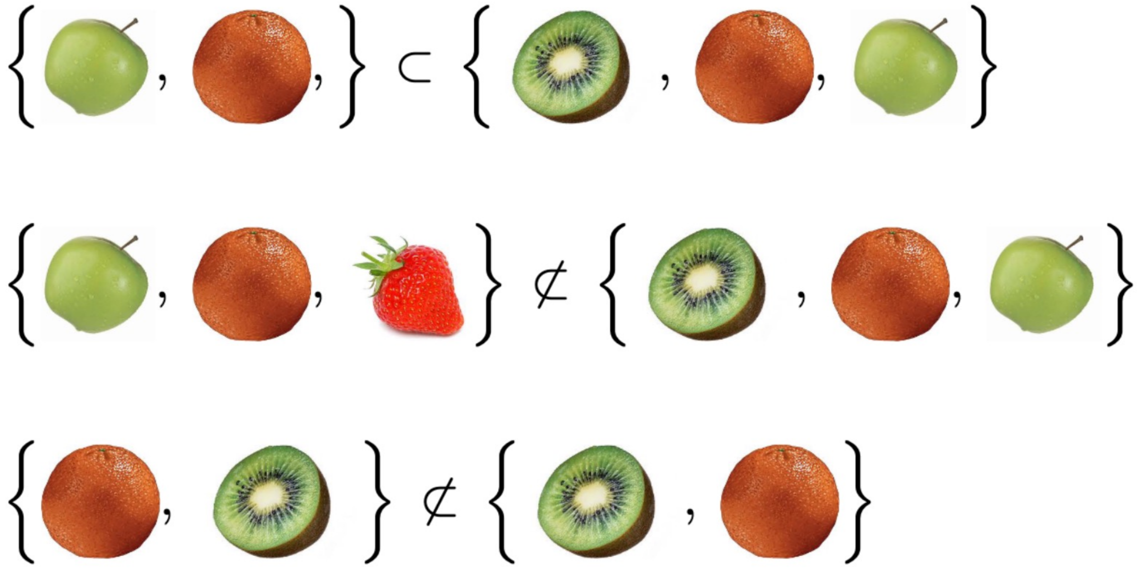
T ⊆ *S* is an assertion that *T* is a subset of *S*. *T* is a subset of *S* iff every element of *T* is also an element of *S*.





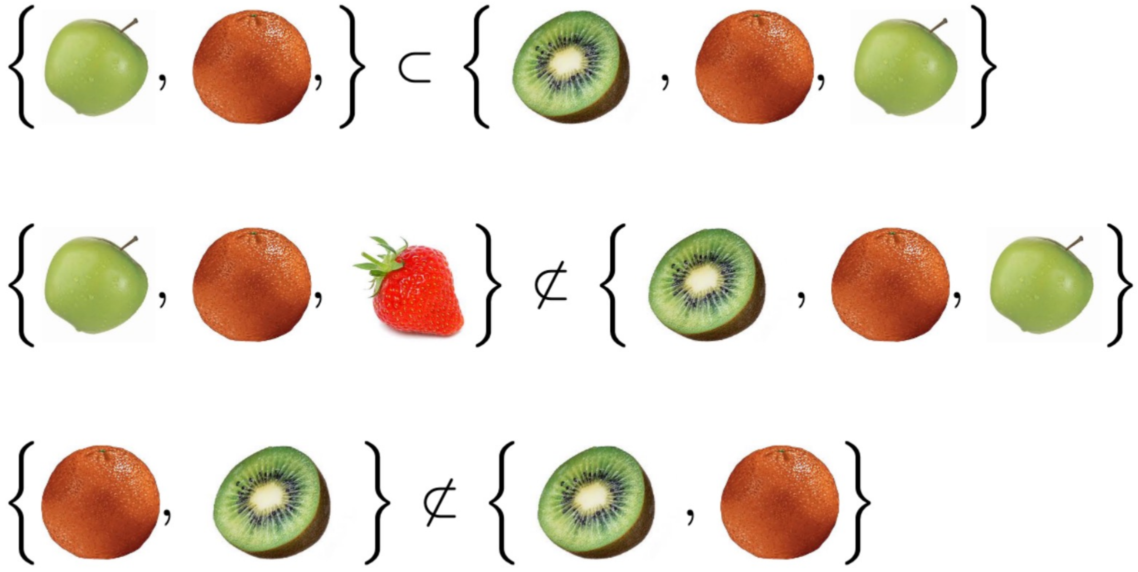
Set Theory: Subset

$T \subset S$ is an assertion that T is a <u>strict</u> subset of S. T is a subset of S iff every element of T is also an element of S <u>and</u> $T \neq S$.



Set Theory: Subset

$T \subset S$ is an assertion that T is a strict subset of S. T is a subset of S iff every element of T is also an element of S and $T \neq S$.



Set Theory: Extensional vs. Intensional

Examples so far: *enumerated* set elements = *extensionally-defined*

Can also define sets in terms of their properties = *intensionally-defined*

- Even numbers are defined to be divisible by two

- Chairs are furniture that you can sit on

Write on board

Relationship between Ontologies and Sets

Are categories sets?

Why or why not?

How would you decide?

Agenda

- Knowledge Representation: Motivation
- Ontologies: Categories and Relations
- Background Review: Set Theory Notation
- **Clicker Questions**
- *Bonus*: OOP as objects in an ontology

Which of the following is *not* an ontology?

Feel free to ask clarifying questions about each of these

A) Dewey Decimal System (library book categorization by topic)

B) Binomial nomenclature (scientific name for living things)

C) Computer Science major requirements

D) Phylogenic Trees

Agenda

- Knowledge Representation: Motivation
- Ontologies: Categories and Relations
- Background Review: Set Theory Notation
- Clicker Questions
- ***Bonus*: OOP as objects in an ontology**

Concretizing Ontologies: Object-Oriented Programming

Write on board

Monday: Predicate Logic

Gluing it together before we break it apart…