Introduction to Intelligent Systems

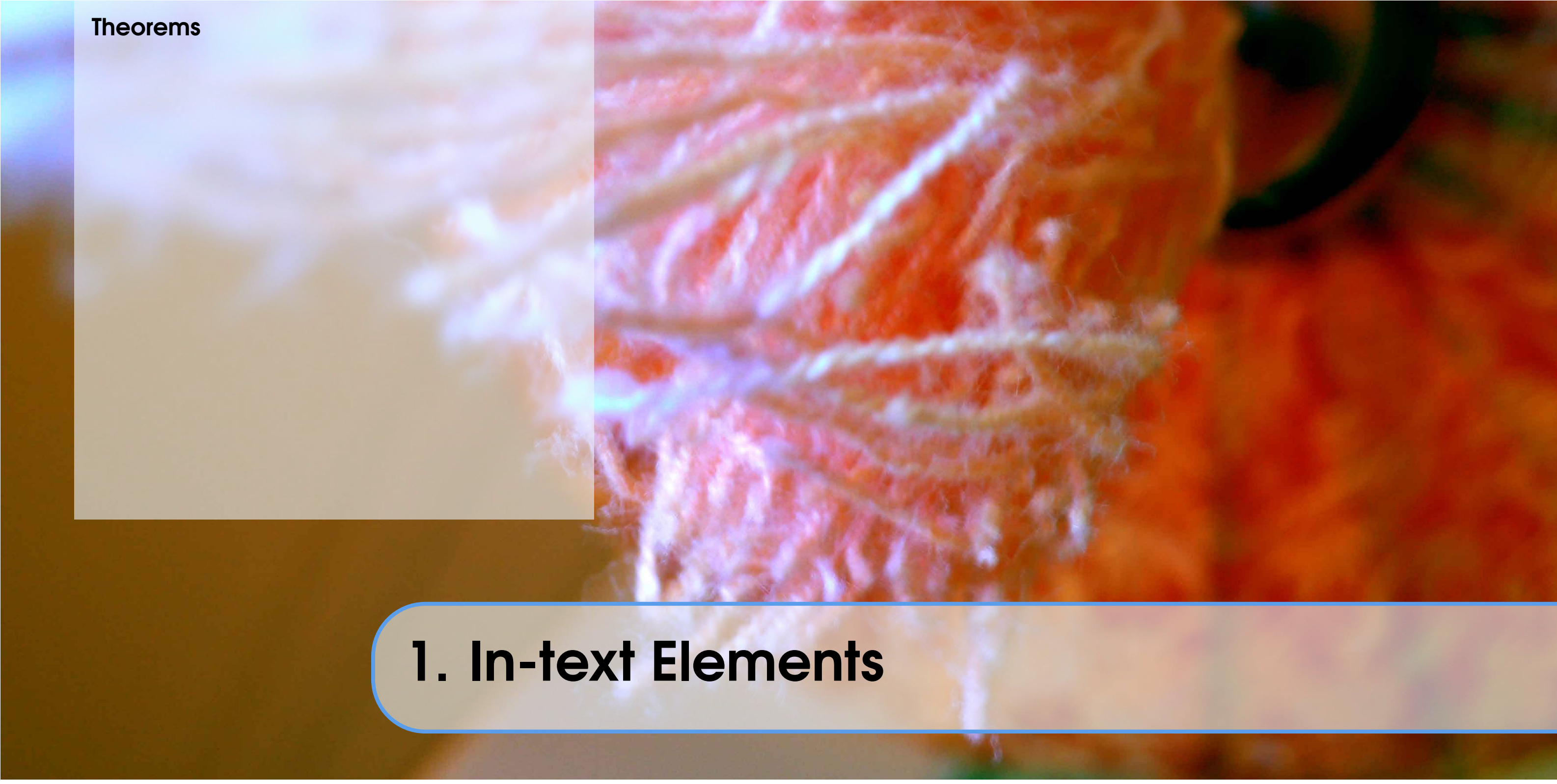
Stefan Klaus

NOT PUBLISHED

GITHUB.COM/DARKLORDNORGE

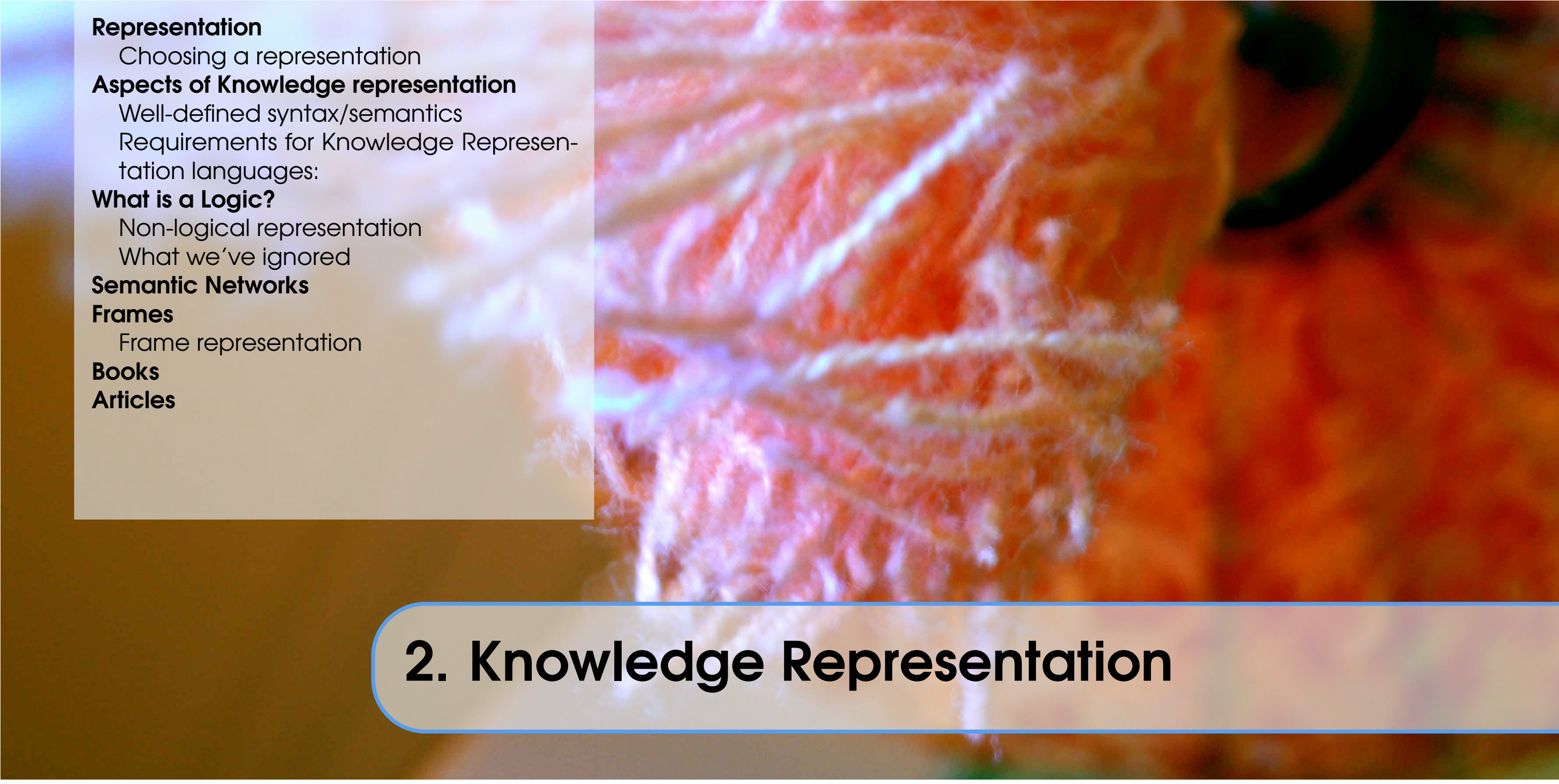
First printing, March 2013

Contents



1. In-text Elements

1.1 Theorems



2. Knowledge Representation

2.1 Representation

AI agents deal with knowledge(data).

- Facts(believe & observe knowledge)
- Procedures("how to" knowledge)
- Meaning (relate & define knowledge)

Right representation is crucial

- Early realisation in AI
- Wrong choice can lead to project failure
- Active research area

2.1.1 Choosing a representation

For certain problem solving techniques.

- Best representation already known
- Often a requirement of the technique
- Or a requirement of the programming language(e.g. Prolog)

Examples:
First order theorem proving(first order logic), Inductive logic programming(logic programs) and Neural network learning(neural networks).

Some of the general representation schemes are suitable for many different(and new) AI applications.

Representation of:

- Declarative knowledge(what, objects, structure)
- Procedural knowledge(how, actions, performance)

Representation formalisms

- Declarative knowledge
  - Frames, Semantic Networks, Inheritance Hierarchies, Schemata,...
- Procedural knowledge
  - Algorithms, Procedures, Planes, Rules,...

Declarative examples

Information about items in a store:

```
cheaper(coca_cola, pepsi)
tastier(coca_cola, pepsi)

if(cheaper(x, y) && (tastier(x, y))
        then but(x)
```

Procedural example

Shopping script:

1. Make a list of all items to buy
2. Walk to the shop
3. For each item on the list, get the item and add it to the shopping
4. Walk to the checkout counter
5. Pack the items
6. Pay
7. Walk home

2.2 Aspects of Knowledge representation

Syntax:

- Possible (allowed) constructions
- For example: colour(my_car, red), my_car(red), red(my_car), etc.

Semantics:

- What the representation **means** (and how it maps to the real world)
- Example:
  - Colour(my_car, red) means: "my car is red", "paint my car red", etc.

Inference:

- The interpreter
- Decides what kind of conclusions can be drawn
- For example: Modus ponens($P, P \rightarrow Q, therefore Q$)

2.2.1 Well-defined syntax/semantics

Knowledge representation languages should have precise syntax and semantics.
You must know exactly what an expression means in terms of object in the real world.

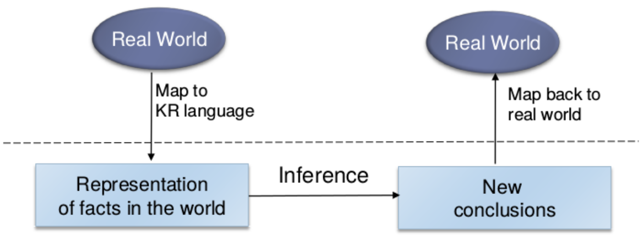


Figure 2.1: Knowledge representation syntax/semantics

2.2.2 Requirements for Knowledge Representation languages:

Representation adequacy:

- Should allow for representing all the required knowledge

Infernal adequacy:

- Should allow the inference of new knowledge

Inferential efficiency:

- Inferring new knowledge should be efficient

Clear syntax and semantics:

- Unambiguous and well-defined

Neutralness:

- Easy to read and use

2.3 What is a Logic?

A language with concrete rules.
No ambiguity in representation, however there may be errors. Allows for unambiguous communication and processing.
Is very unlike natural languages like e.g. English.
Many ways to translate between languages a statement can be represented in different logics, and possibly differently in same logic.

It should not be confused with logical reasoning, logic are languages, reasoning is a process(which may use logic).

2.3.1 Non-logical representation

Logic representation have restrictions and can be hard to work with. Many AI researches searched for better representations.

Non-logical:

- Semantic networks
- Conceptual graphs
- Frames
- Scripts
- Production rules
- ...

2.3.2 What we've ignored

Objects in the world tend to be related to each other.

- Classes, superclasses & subclasses, part / whole hierarchies
- Properties are *inherited* across relationships

The state of the world can change over time.

- Explicit representation of time
- Frame problem: representing the effects of action in logic without having to represent explicitly a large number of intuitive obvious non-effects
- Non-monotonic reasoning

We must reason without complete knowledge

- Closed world assumption

Not all knowledge is "black & white":

- Uncertainty, statistics, fuzzy logic,...

Defaults and exceptions:
Exception for a single object, a property of the object must be set to the (exception) value.

2.4 Semantic Networks

Semantic networks are essentially a generalization of inheritance hierarchies.
Each node is an object, class, concept, or event.

Each link is a relationship.
- is-a (the usual sublcass or element relationship
- has-part or part-of
- any other relationship that makes sense in context(e.g. owns thing x)

Semantic networks represent knowledge as a network or graph(easily stored on the computer).
By traversing the network we can find:
- Elephant x likes apples(by inheritance)
- That certain concepts related in certain ways(e.g. apples and elephants)

2.5 Frames

Devised by Marvin Minsky in 1974.
Is an extension to semantic networks, and incorporates certain valuable human thinking characteristics:
Expectations, assumptions, stereotypes, Exceptions, Fuzzy boundaries between classes.

Frames often allowed you to say which things were just typical(*) if a class, and which are definitional, so couldn't be overridden.
Frames also allow multiple inheritance(Nellie is an Elephant AND a circus animal).

2.5.1 Frame representation

Bibliography

Books

Articles

Index