Great Moments in Knowledge Representation:
The 1984 Complexity Convergence

Ron Brachman
Yahoo! Labs

Hector Levesque
University of Toronto

September 14, 2010
“It may be speculated that a large part of human thought consists of manipulating words according to rules of reasoning and rules of conjecture. From this point of view, forming a generalization consists of admitting a new word and some rules whereby sentences containing it imply and are implied by others.”

“This idea has never been very precisely formulated nor have examples been worked out.”
The Fundamental Idea

- A good part of thought involves a store of represented beliefs and procedures that operate on them...to ‘produce’ new beliefs...that impact (decisions on) what to do.

- This was a radical idea that put the focus on the knowledge – what can be known, and how, and what follows from what is known.

Knowledge - its representation and procedures for manipulating it - becomes a first-class object of study.

Actions are conditioned on what is believed.

The “knowledge-based” approach to AI.
Knowledge Representation and Reasoning

- Over the next quarter century, starting with classical logic, a wide variety of formalisms for representing knowledge were developed and used.
The 1984 Convergence

- Today we will focus on an important turning point in the field in 1984
  - “The Tractability of Subsumption in Frame-Based Description Languages” (AAAI-84)

But this was not just a single “bolt from the blue” idea

It was built on several key trends that crystallized in the late ‘70’s and early ‘80’s to make an important set of observations and a new kind of analysis possible
Today’s Agenda

- Early KR
- Key Threads
- The 1984 Result
- The Aftermath
The Paper Trail

KL-ONE Overview

1981 KL-ONE Workshop

Functional Approach

Competence (AAAI-82)

Krypton (AAAI-83)

The Tractability of Subsumption in Frame-Based Description Languages

Ronald J. Brachman
Rector J. Levesque
Fudan University Laboratory for Artificial Intelligence Research
501 Alice Avenue
Palo Alto, California 94304

ABSTRACT

A knowledge representation system is an important part in the construction of a knowledge-based program. However, the requirement for the programing of such a system is not always straightforward. This paper presents a new approach to the construction of the system, called the Functional Approach. The approach is based on the idea that the system should be constructed from a set of simple, well-defined modules. Each module is responsible for a specific aspect of the system, such as the representation of concepts, the representation of relations, the representation of actions, the representation of time, etc. The modules are then combined to form the complete system.

1. Introduction

There are many different ways of constructing knowledge representation systems. This paper presents a new approach to the construction of such systems. The approach is based on the idea that the system should be constructed from a set of simple, well-defined modules. Each module is responsible for a specific aspect of the system, such as the representation of concepts, the representation of relations, the representation of actions, the representation of time, etc. The modules are then combined to form the complete system.

2. Subsumption in Frame Languages

Among the most popular knowledge representation languages used today is the computer language, for example, FLANN [2], and RuleML [3]. Frames are the ability to define structured types, typically a frame contains a set of general frames (the superframe) and a set of instances (the subframes). The superframe contains a set of slots (the attributes) of the subframe. The slots contain the value of the subframe for all instances of the frame. The subframe is used as a predicate that all instances of the frame must satisfy (i.e., the value may not be a default). It is a simple way to define a constraint, which can be an individual value, a variable, or a structure value.

3. Conclusion

In conclusion, this paper presents a new approach to the construction of knowledge representation systems. The approach is based on the idea that the system should be constructed from a set of simple, well-defined modules. Each module is responsible for a specific aspect of the system, such as the representation of concepts, the representation of relations, the representation of actions, the representation of time, etc. The modules are then combined to form the complete system.

References


Acknowledgments

This work was supported by the National Science Foundation under Grant IRI-8502995.

Keywords: knowledge representation, frames, subsumption, modularity.
Antediluvian KR

- An important early focus came from memory models
  - Quillian’s thesis, Collins and Quillian (1960’s)
  
- Network models became very popular
  - Reflected associativity amongst words/concepts
  - Path-based inferences reflected human psychology experiments, intuition
Characterization was almost exclusively in the form of diagrams

- Syntax focused on naming diagram structures (not compositional expressions)
- Path-based inferences were based solely on symbol-level structure

In many cases “semantics” simply reflected what an implemented system did
- Not functional, nothing like standard formal semantics
How Powerful Were Semantic Nets?

(conceived by B.C. Smith)
Key Threads

- Clarity of meaning of representations
  - Terminology and assertions
  - Subsumption as focal inference
  - Linear syntax
  - Formal semantics
  - Functional approaches to KR
  - Assessment of complexity
What’s in a Link?

- As semantic nets grew popular in the ’70’s, their diversity exploded
  - Links were typically labeled with English words, subject to wide interpretation
  - “IS-A” had many meanings
- Attempts to clarify meaning
  - In some cases semantics captured by simple conventional translation to FOL (Arguably, more convenient syntax)
- Woods’ “What’s in a Link” paper (1975)
  - Multiple interpretations of different links with same names
  - Conflation of storing factual knowledge and modeling associative connections
  - Intensions (“the morning star” vs. “the evening star”)

“What’s in a Link” (1975)
What’s in a Concept?

- Brachman’s thesis (1977): considered a lower level of representation with a set of primitive operators for constructing concepts
  - Avoided ambiguity, allowed consistent (semi-)formal meaning
  - “epistemological primitives”: could look at adequacy of concept-forming operators independent of intuition and gratuitous meaning of NL words
KL-One

- Implementations of Brachman’s “SI-Net” notation at BBN and ISI
  - Used in multiple applications
  - Inspired much discussion of foundations of representation of concepts and objects
  - Broader context included innovative and thoughtful work on related ideas in by Brian Smith, Stu Shapiro, Len Schubert, the KRL team (PARC), and others
- First KL-One workshop: 1980
- Key inference: classification
  - Automatic placement of concepts in concept taxonomy
    - Based on meaning of structures rather than user whim
  - Based on central notion of concept (description) subsumption

KL-ONE Overview

Brian Smith  Jim Schmolze
me  David Israel
Key Threads

- Clarity of meaning of representations
- **Terminology and assertions**
  - Subsumption as focal inference
  - Linear syntax
  - Formal semantics
  - Functional approaches to KR
  - Assessment of complexity
Terminological Logics

- KL-One inspired a focus on the distinction between terminology and assertion
  - The elements out of which assertions can be constructed vs. asserting beliefs themselves
- Management of technical vocabularies
  - Compound predicates (e.g., “father-of-only-sons”) whose meaning is dictated by their parts
  - Operators for forming compound predicates
  - Classification was strictly a terminological inference
    - Contingent statements not taken into account
  - Focus on descriptions and their structure
Saying Things about the World

- “Nexuses” (in contexts) as first stab in KL-One
  - More of a wart on the description language
  - “Description wires” – still operating on diagrams?
- Best method for saying things about the world?
  - Expressiveness needed to be appropriate
    - Incompleteness (including disjunction)
    - Defaults, knowledge about knowledge (Levesque thesis)
- 1981 KL-One workshop
  - Birth of the “TBox”/”ABox” distinction
  - Major break in community
  - Significant clarity around import of various components of KR
- “Competence” paper
  - Different standards of adequacy for assertion language than term-structuring language

KL-ONE Overview

1981 KL-ONE Workshop

Competence (AAAI-82)
Key Threads

- Clarity of meaning of representations
- Terminology and assertions
- **Subsumption as focal inference**
- Linear syntax
- Formal semantics
- Functional approaches to KR
- Assessment of complexity
Hybrid Systems

- Could gloss KL-One-type structures in FOL
  - Some hybrid systems created FOL axioms for terminology and integrated with purely assertional FOL statements
- *Essential* integration: different frameworks for different purposes
  - Terms in TBox *are* the non-logical symbols in the ABox
  - Explored in Krypton (with Fikes, others)
  - Stickel’s Theory Resolution for integration

- Subsumption (and disjointness) computation is critical
  - Theory Resolution: complex unification + resolution – can’t compute the intended augmented inferences in the ABox without subsumption, disjointness
Key Threads

- Clarity of meaning of representations
- Terminology and assertions
- Subsumption as focal inference
- **Linear syntax**
  - Formal semantics
  - Functional approaches to KR
  - Assessment of complexity
The Evolution of Syntax

- Pre-1982: Tbox understood as graphical structure
  - “Competence in Knowledge Representation” (AAAI-82)
    - TBox understood as repository for terms
    - characterized by noun-phrase like descriptions

\[
\text{transistor} \leftarrow (\text{PrimGeneric} (\text{NRGeneric} \text{ device connection 3 3}))
\]

= “a transistor is, among other things, a device with at least 3 and at most 3 connections”
Linear Syntax

- **AAAI-83 Krypton paper:**
  - ABox sentences based on usual logical connectives
  - TBox phrases derived from KL-ONE operators
  - family of languages derive from choice of operators

\[
\text{(VRGeneric} \ (\text{ConjGeneric} \text{ unmarried-person man}) \\
\text{(VRDiffRole} \text{ sibling man}) \\
\text{(NRGeneric} \text{ person (RoleChain} \text{ child child} \ 1 \ \infty))
\]

= a bachelor whose brothers have grandchildren

- **1984 paper:** \(FL\) and \(FL-\), two very simple TBox languages

\[
\text{(AND} \text{ person} \\
\text{(SOME} \text{ child}) \\
\text{(ALL} \text{ (RESTR} \text{ child male) lawyer}) \\
\text{(ALL} \text{ (RESTR} \text{ child female) doctor})
\]
Key Threads

- Clarity of meaning of representations
- Terminology and assertions
- Subsumption as focal inference
- Linear syntax
- **Formal semantics**
- Functional approaches to KR
- Assessment of complexity
The Evolution of Semantics

- Pre-1982: Complex semantics, partly procedural
- 1982-83: TBox terms translated into meaning postulates in first-order logic

\[ P \leftrightarrow [\text{ConjGeneric} \ [\text{VRGeneric friend rich}] \\
\text{VRGeneric child [VRGeneric friend happy]]}] \]

\[ \forall x. P(x) \iff \forall y (\text{Friend}(x,y) \Rightarrow \text{Rich}(y)) \land \\
\forall y (\text{Child}(x,y) \Rightarrow \forall z (\text{Friend}(y,z) \Rightarrow \text{Happy}(z))) \]
Truth-Functional Semantics for TBox

- 1984 paper: Truth-functional semantics inspired by Tarski
  - concepts like 1-place predicates
  - roles like 2-place predicates
  - each non-atomic concept/role gets an extension determined by the term operator used, e.g.,
    \[ \mathcal{E}[(\text{ALL } r \ c)] = \{ x \mid \text{if } <x, y> \in \mathcal{E}[r], \text{ then } y \in \mathcal{E}[c] \} \]

- Subsumption is the TBox version of logical entailment
  - in all interpretations, the extension of \( P \) is a subset of the extension of \( Q \)
Key Threads

- Clarity of meaning of representations
- Terminology and assertions
- Subsumption as focal inference
- Linear syntax
- Formal semantics
- Functional approaches to KR
- Assessment of complexity
Functional Interfaces

- Pre-1982: Service from KR system to AI programs
  - ABox: automated theorem prover = compute logical entailment
  - TBox: access to implementation data structures + some specialized operations (classification)
- Levesque's PhD thesis (1981) proposes TELL and ASK for ABox
  - TELL: add new facts
  - ASK: determine what is known
    - not just logical entailment because of negative information (used in defaults)
    - Implementation structures not relevant to or accessible by user
- 1983: Krypton built along these lines
- 1984 paper: subsumption ASK only

Krypton (AAAI-83)
Tractability (AAAI-84)
Subsumption as a Service

- KR component of intelligent system providing a service
  - Manages knowledge, performs inferences
  - User has no access to data structures, KR component is purely about knowledge
- Subsumption as key inference
  - In hybrid system, subsumption and disjointness in TBox are critical to extending ABox service to include non-logical symbols honoring meaning of terminology (extension of resolution)
- Responsiveness of TBox subsumption service to ABox requests is critical to overall performance
  - ABox reasoner accesses TBox between deduction steps
  - Specialized “frame”-like inferences should be simple
  - Otherwise, just use meaning postulates and general theorem-proving
Key Threads

- Clarity of meaning of representations
- Terminology and assertions
- Subsumption as focal inference
- Linear syntax
- Formal semantics
- Functional approaches to KR
- Assessment of complexity
Complexity of Reasoning

- 1982: have well-defined operation (subsumption) over linear expressions with simple truth-functional semantics
- Pre-1983: difficulty in implementing efficient TBox
  - attempts without VRDiff work fine
  - many attempts with VRDiff found to be buggy

  
  (ConjGeneric  Person
   (VRGeneric  child  (ConjGeneric  female  student)))
  "a person whose children are all female students"

  
  (ConjGeneric  Person
   (VRGeneric  (VRDiffRole  child  female)  student)))
  "a person whose female children are all students"

- 1983: discover an encoding of CNF formulas into TBox terms (without negation)
Pre-1984 Complexity Results in AI

<this page intentionally blank>
The Tractability of Subsumption in Frame-Based Description Languages

Ronald J. Brachman
Hector J. Levesque
Fairchild Laboratory for Artificial Intelligence Research
4001 Miranda Avenue
Palo Alto, California 94304

ABSTRACT

A knowledge representation system provides an important service to the user of a knowledge-based system if it can operate automatically and with the required quality of knowledge. This knowledge system must be tractable in terms of both its operational quality (i.e., in diagnosis, planning, and reasoning), and its computational tractability if it is to be useful.

There are many different styles of knowledge representation systems, but the one that is most common is the frame-based system (see [2] for an overview of this tradeoff). In this brief paper, we present a formal analysis of the computational complexity of frame-based systems. We illustrate how good representation systems can be used in the design of a representation system, even when our initial ideas about the language itself are not adequate. It is true that, even an apparently good representation language can prove intractable.

1. Introduction

There are many different styles of knowledge representation systems, but they all have at least this in common: the representation system is supposed to provide both a battery of logical formulas for the knowledge of the knowledge-based system in which it is embedded, as well as a set of automatic inference rules over these formulas. Typical inference systems are fully automated by AI representation systems include inheritance of properties, set membership, and set inclusion, predicate inference, type subsumption, and resolution.

Here we address a fundamental problem in the nature of the service to be provided by knowledge representation systems: the greater the expressiveness of the language of representing knowledge, the better it becomes to compute the inferred formulas (see [2] for an overview of this tradeoff). In this brief paper, we present a formal analysis of the computational complexity of frame-based systems. We illustrate how good representation systems can be used to design a representation system, even when our initial ideas about the language itself are not adequate. It is true that, even an apparently good representation language can prove intractable.

2. Subsumption in Frame Languages

Among the most popular representation languages we use today are those based on the notion of frame (e.g., C. [3], [6], and [7]). Frames give us the ability to define structured types; typically a frame comprises a set of more general frames (i.e., superframes) as well as a set of descriptions of the attributes (slots) of instances of the frame. The most common type of description specifies a restriction on the value of the slot of the all instances of the frame. The restriction can be as specific as a particular value that all instances of the frame must exhibit (alternatively, the value may be just a default, in which case an individual instance the values provided by the default slot).

The complexity of subsumption in frame-based systems is in which case this value restriction is usually a pointer to another frame. Less commonly, the number of required frames is also specified in a slot restriction (often in terms of a minimum and a maximum number of attribute values). The generalization relation between frames is then a generic one and a frame is a type of another, explicitly formed a taxonomy, or inheritance directly.

Notationally, a frame might be defined by a list of superframes (with either an explicit or implicit "of instance" relation [3]), followed by a set of slot restrictions expressed by attribute/value-description pairs (with attributes and value-description usually separated by a colon). For example, the simple frame

```
[FRAMES
  [FAMILY (2, 1)]
  [son-kin] LARRY
  [daughter] DIANA
]
```

is intended to be a structured type representing the concept of a woman that has at least one child and all of whose sons (i.e., male children) are lawyers and all of whose daughters are doctors. Similarly, the more complicated frame

```
[FRAMES
  [STUDENT FEMALE
    department: COMPUTER-SCIENCE
    enrolled: 3-2-1]
  [MATH DEPARTMENT: ENGLISH-DEPARTMENT]
]
```

is intended to be a structured type that describes female Computer Science students taking at least three graduate courses in a department within a school of Engineering.

There is a natural correspondence between this frame form of description and usual phrases in natural language. For example, the above frames might just as well have been written as "student and a female whose department is computer-science, and who has at least 3 enrolled-courses, each of which is a graduate-course whose department is an engineering-department." A single set of translation rules would allow us to more easily both frame form to (almost) readable English.

```
[FRAMES
  [FAMILY (2, 1)]
  [son-kin] LARRY
  [daughter] DIANA
]
```

While the use of natural restrictions is not widespread, they have been used extensively in KL-ONE [8] and languages like [4]. They seem to be a useful generalization of the conceived meaning of expressions like "male" to include those who are not male.

Finally, frames are given names as well, for example, we might also label our frame

```
[FRAMES
  [FAMILY (2, 1)]
  [son-kin] LARRY
  [daughter] DIANA
]
```

as "family". As a result, a person's name could become a conjunction of frames ("student and a female") and a slot that had one can fill might translate into a single "whose" clause ("whose department is computer-science"). And a slot
The 1984 Result: A Tale of Two Languages

- $\mathcal{FL}$:
  - Concepts
    - Atomic concepts: $A$
    - Concept intersection: $(\text{AND } C \text{ D})$
    - Limited existential quantification: $(\text{EXISTS } r)$
    - Value restriction: $(\text{ALL } r C)$
  - Roles
    - Atomic role name: $r$
    - Role restriction: $(\text{RESTR } r C)$

- $\mathcal{FL}$-:
  - Same as $\mathcal{FL}$ without role restriction

A new type of AI technical result!

Subsumption in $\mathcal{FL}$ is Co-NP-Hard
(reduction of tautology checking in 3DNF to subsumption)

Subsumption in $\mathcal{FL}$- can be decided by a structural algorithm, with polynomial-time complexity
Implications

- The “computational cliff”
  - Truly dramatic change in complexity from what at first looks like a small addition to the language

- In general, there is a fundamental tradeoff between expressiveness and complexity in KR frameworks
The Aftermath

- Explosion of derivative results
  - 100’s of papers
- Drastic improvement in clarity about meaning of representations and adequacy of reasoning
- Technical rigor of results
Before

REFERENCES


After

2.6.2 Partition

In order to...e 3.

2.3.

Theorem 2: (b) The P problem is NP-complete.

Theorem 3: Con...is studied in...eorem 4:...he...heorem 5.

Lists of roles are interpreted as the composition of relations:

\[ R(A_1, \ldots, A_n) \rightarrow \bigwedge_{i=1}^{n} R_i(A_i) \]

We interpret defined concepts as subsets of M as follows:

- \( C \ominus D \in P \)
- \( C \sqcap D \in P \)
- \( C \cup D \in P \)
- \( C \setminus D \in P \)

Given a list of \( e \) and \( d \), the semantics can be used to determine the element of the concept.

It is shown that the KL-ONE system is equivalent to the NTKL-expression (and all \( \{ C \setminus D \} \}

Subsumption and Consistency are defined with respect to:

- A concept expression \( C \) subsumes a concept expression \( D \) if for all interpretations \( I \), we have \( I(C) \supseteq I(D) \).
- A concept is consistent, if there exists an interpretation \( I \) such that \( I(C) \neq \emptyset \), otherwise \( C \) is called inconsistent.
- Two concepts \( C \) and \( D \) are equivalent, if \( C \equiv D \) and \( D \equiv C \).

Since we have no concept of \( \aleph \), and since the empty concept can be defined, subsumption problems in NTKL is equivalent to inconsistency problems.

2.1. Lemma. Let \( C \) be a concept expression with respect to NTKL. Then the following three statements are equivalent:

i) \( D \) subsumes \( C \).
ii) \(-D \equiv C \) is an inconsistent concept.
iii) \(-D \) is subsumed by \(-D \) (the empty concept).

Proof. The concept \( D \) subsumes \( C \) if and only if \( I(C) \subseteq I(D) \). This is true if and only if \( I(D) \setminus I(C) = \emptyset \). Thus, \( I(D) \setminus I(C) = \emptyset \) is equivalent to \( I(D) \) being a consistent concept. Obviously, \( I(D) \setminus I(C) = \emptyset \) for all interpretations \( I \) (hence \( I(C) \) is equivalent to \( I(C) \).

Subsumption is not equivalent to consistency in the language of NTKL, since \( \aleph \) all concepts are consistent. Even if \( \aleph \) and the role-defining operator \( (\bullet \circ C) \) from NTKL are permitted, then all concepts remain consistent, which follows by considering an additional element \( \exists (M) \) with \( M = \{m\} \), where all concept symbols are interpreted as \( M \) and all roles as \( \{m, m\} \). In this interpretation, the restriction of a role has no effect, and all defined concepts have \( M \) in extension. Thus all concepts remain consistent.

Let us compare the expressiveness of NTKL and NTKL with the following knowledge representation languages.

Obviously, NTKL is a sublanguage of KL-One. NTKL has less expressiveness than the sublanguage of NIKL used by P.P. and C. in their work. To show undecidability of subsumption in NTKL, the ALC concept expression is equivalent to \( (\forall R \bullet \{ R \_C \_R \}) \) in NIKL and to \( (\forall R \bullet \{ R \_C \_R \}) \) in KL. (Levesque and Brachman, 1989). A concept description using role value maps \( (R_1: \ldots, R_n) \rightarrow (S_1, \ldots, S_m) \) is equivalent to the NTKL-expression (and all \( \{ c \_R \} \) \( \{ C \_R \} \) \( \{ S \_R \} \) \( \{ \emptyset \} \) \( \{ \emptyset \} \). I conjecture that the expressiveness of NTKL is strictly smaller, since it is not possible to define functional roles in NTKL.

There is a close relationship between ALC and feature terms. ALC can be seen as feature terms (without negation and union) with the semantics of features changed from (partial) functional to arbitrary binary relations. The same relationship holds between ALC and feature terms with complements and union. Of course, the descriptive power is not really comparable, since for the rules for computing with feature terms and concept expressions in ALC are different.

For example, the rules for computing with feature terms and concept expressions in ALC are different. For example, the rules for computing with feature terms and concept expressions in ALC are different. For example, the rules for computing with feature terms and concept expressions in ALC are different. For example, the rules for computing with feature terms and concept expressions in ALC are different. For example, the rules for computing with feature terms and concept expressions in ALC are different.
Table 1: The complexity of \( M^a \).

<table>
<thead>
<tr>
<th>( M^a )</th>
<th>( A )</th>
<th>( A )</th>
<th>( A )</th>
<th>( A )</th>
<th>( A )</th>
<th>( A )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
</tr>
<tr>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
</tr>
<tr>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
</tr>
<tr>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
<td>( A )</td>
</tr>
</tbody>
</table>

Note: This table shows the complexity of \( M^a \) for different values of \( A \).

Example 7: Consider the case where \( A = \{1, 2, 3\} \) and \( M^a \) is defined as follows:

- \( A \) contains the elements \( 1, 2, 3 \).
- \( M^a \) is defined as a set of rules that determine the complexity of \( A \).

In this case, the complexity of \( M^a \) is \( \mathcal{O}(n^3) \), where \( n \) is the number of elements in \( A \).

Theorem 8 (Domain Cont.)

Let \( A \) be a domain and \( M^a \) be a set of rules. Suppose \( M^a \) contains the following rules:

- Rule 1: \( A \subseteq \mathcal{O}(n^3) \)
- Rule 2: \( A \subseteq \mathcal{O}(n^2) \)
- Rule 3: \( A \subseteq \mathcal{O}(n) \)

Then the complexity of \( M^a \) is \( \mathcal{O}(n^3) \).

Proof: By the definition of complexity, if \( M^a \) contains the rules above, then the complexity of \( M^a \) is \( \mathcal{O}(n^3) \).

Theorem 9 (Domain Cont.)

Let \( A \) be a domain and \( M^a \) be a set of rules. Suppose \( M^a \) contains the following rules:

- Rule 1: \( A \subseteq \mathcal{O}(n^3) \)
- Rule 2: \( A \subseteq \mathcal{O}(n^2) \)
- Rule 3: \( A \subseteq \mathcal{O}(n) \)

Then the complexity of \( M^a \) is \( \mathcal{O}(n^3) \).

Proof: By the definition of complexity, if \( M^a \) contains the rules above, then the complexity of \( M^a \) is \( \mathcal{O}(n^3) \).

Remark 2 (Domain Cont.)

Let \( A \) be a domain and \( M^a \) be a set of rules. Suppose \( M^a \) contains the following rules:

- Rule 1: \( A \subseteq \mathcal{O}(n^3) \)
- Rule 2: \( A \subseteq \mathcal{O}(n^2) \)
- Rule 3: \( A \subseteq \mathcal{O}(n) \)

Then the complexity of \( M^a \) is \( \mathcal{O}(n^3) \).

Proof: By the definition of complexity, if \( M^a \) contains the rules above, then the complexity of \( M^a \) is \( \mathcal{O}(n^3) \).

Proposition 3 (Domain Cont.)

Let \( A \) be a domain and \( M^a \) be a set of rules. Suppose \( M^a \) contains the following rules:

- Rule 1: \( A \subseteq \mathcal{O}(n^3) \)
- Rule 2: \( A \subseteq \mathcal{O}(n^2) \)
- Rule 3: \( A \subseteq \mathcal{O}(n) \)

Then the complexity of \( M^a \) is \( \mathcal{O}(n^3) \).

Proof: By the definition of complexity, if \( M^a \) contains the rules above, then the complexity of \( M^a \) is \( \mathcal{O}(n^3) \).

Corollary 4 (Domain Cont.)

Let \( A \) be a domain and \( M^a \) be a set of rules. Suppose \( M^a \) contains the following rules:

- Rule 1: \( A \subseteq \mathcal{O}(n^3) \)
- Rule 2: \( A \subseteq \mathcal{O}(n^2) \)
- Rule 3: \( A \subseteq \mathcal{O}(n) \)

Then the complexity of \( M^a \) is \( \mathcal{O}(n^3) \).

Proof: By the definition of complexity, if \( M^a \) contains the rules above, then the complexity of \( M^a \) is \( \mathcal{O}(n^3) \).

Proposition 5 (Domain Cont.)

Let \( A \) be a domain and \( M^a \) be a set of rules. Suppose \( M^a \) contains the following rules:

- Rule 1: \( A \subseteq \mathcal{O}(n^3) \)
- Rule 2: \( A \subseteq \mathcal{O}(n^2) \)
- Rule 3: \( A \subseteq \mathcal{O}(n) \)

Then the complexity of \( M^a \) is \( \mathcal{O}(n^3) \).

Proof: By the definition of complexity, if \( M^a \) contains the rules above, then the complexity of \( M^a \) is \( \mathcal{O}(n^3) \).
Description Logics

- Many results evaluating variants of basic logic
- CLASSIC
  - Practical implementation directly influenced by complexity results
  - Significant synergy between implementation and theory
- Many others
  - Exploration of highly expressive but effective systems
  - Evolution of inference methods
    - Tableaux
- Simpler syntax
- OWL

\[
\text{Conjunction: } C \sqcap D \\
\text{Value restriction: } \forall r \\
\text{Existential quantification: } \exists r \\
\text{Role restriction: } r|_c
\]

...
To Ponder

- Did the field go overboard?
- Worst-case vs. average-case analysis
- Still too much focus on TBoxes?
  - “Ontologies”
Recap

- Multiple threads came together in the mid-80’s to change the field
  - Clarity of meaning of representations
    - Carefully designed description-forming operators
  - Terminology/description structure
    - TBox/ABox distinction
  - Subsumption as focal inference
    - Critical to support essential hybrid KR systems
  - Linear syntax
  - Formal semantics
  - Functional approaches to KR (ASK/TELL)
  - Apparatus for assessing complexity
  - A fundamental tradeoff between expressiveness and computational tractability

- The effect was dramatic!
Thank you!