AI-KI-B

Introduction to Artificial Intelligence

Ute Schmid & Diedrich Wolter
Practice: Johannes Rabold

Cognitive Systems and Smart Environments
Applied Computer Science, University of Bamberg

last change: 12. Juni 2019, 13:05

Agenda

Topics for today

- Part I: The field of Knowledge Representation (KR)
- Part II: Selected KR approaches
- Part III: Constraint-Based Reasoning

Educational objectives: being able to...

- describe the aims and challenges of KR
- summarise selected approaches: frames, semantic networks
- define the constraint satisfaction problem
- implement simple search techniques to solve CSPs

Quiz!

1. Why can't we apply A* to search for a good move in a two-player adversarial game?
2. What is a characteristic of games for which we don't have super-human AI yet?
3. What are our options to tackle games in which the game tree is too large to traverse?

Part I: KR

Knowledge Representation (KR)

When the system is required to do something that it has not been explicitly told how to do, it must reason[. . .] (Handbook of AI, Vol. 1, 1981)

- Fundamental capability: being able to **reason**, i.e., being able to draw conclusions, to plan, to learn, etc.
- Reasoning processes need to operate on data: a **knowledge base**, an instantiation of a **knowledge representation**
- The field of **Knowledge Representation (KR)** is the field of studying representation techniques that empower effective reasoning processes.

Knowledge

In AI we distinguish **data**, **information**, **knowledge**, and **facts**.

data
: digital representation of any kind, uninterpreted

information
: in the sense of Shannon, abstract and objective

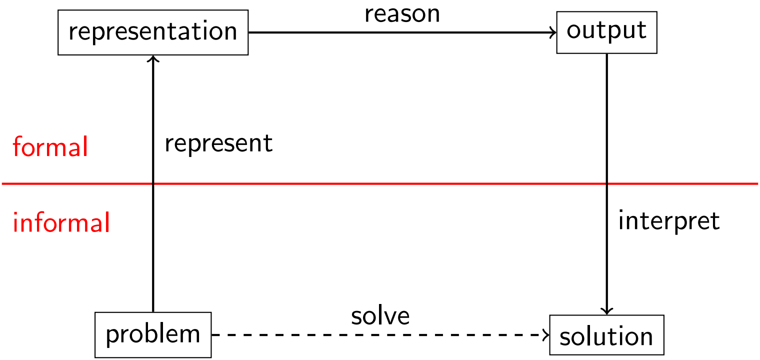
knowledge
: pieces of information that can interpreted and connected with other pieces of information by an agent, subjective to the agent

fact
: a single piece of knowledge

The set of all facts is called the **knowledge base**.

Example: "Ascension Thursday in 2019 is on 30rd of May." is a piece of information. If it gets connected to one's common sense – "No AI lecture next week!" – it becomes knowledge.

Representations are **lossy mappings**, i.e., **not injective**.



$$\text{solve} \overset{!}{\equiv} \text{interpret} \circ \text{reason} \circ \text{represent}$$

$\rightsquigarrow$ inevitable imperfections of any representation restricts range of equivalence

Example

Recall the puzzle from first assignment: getting farmer, fox, goose, and corn across river:

```
(define *start-state* '((farmer fox goose corn) ()))
(define *end-state*   '(() (farmer fox goose corn)))
```

Representation abstracts from boat (that's OK since the boat can only be on the side of the river on which the farmer is), and the process of riding the boat.

For such simple problems, KR techniques do not matter – it get tricky when dealing with more realistic problems.

Kinds of Knowledge I

There are various modalities of knowledge, for example:

- “Hans is sitting on this chair.”
  propositional fact
- “Hans had been sitting on this chair.”
  temporal view on propositional fact
- “Hans could be sitting on this chair.”
  possibilities
- “Jane thinks Hans is sitting on the chair.”
  believe state
- (...)

There are different **domains** of knowledge:

- mathematics, e.g., $x < y \Rightarrow x < y + 1$
- everyday physics, e.g., gravity
- time
- causality
- art
- ...

$\rightsquigarrow$ KR Subfield of **Common Sense** exclusively considers representation of and reasoning with humans' everyday knowledge about the world.

Kinds of Knowledge III

Knowledge can come in various degrees of **(un)certainty**:

- **crisp/certain** knowledge: $2 + 3 = 5$
  (we can assign a binary truth value to statements)

Kinds of Knowledge III

Knowledge can come in various degrees of **(un)certainty**:

- **crisp/certain** knowledge: $2 + 3 = 5$
  (we can assign a binary truth value to statements)
- **uncertain** knowledge

Knowledge can come in various degrees of **(un)certainty**:

- **crisp/certain** knowledge: $2 + 3 = 5$
  (we can assign a binary truth value to statements)
- **uncertain** knowledge
  - **coarse** knowledge: sun sets around 18:00 – 21:00
    (certainly, it will set some time between 18:00 and 21:00)

Knowledge can come in various degrees of **(un)certainty**:

- **crisp/certain** knowledge: $2 + 3 = 5$
  (we can assign a binary truth value to statements)
- **uncertain** knowledge
  - **coarse** knowledge: sun sets around 18:00 – 21:00
    (certainly, it will set some time between 18:00 and 21:00)
  - **vague** knowledge: a soft drink for 4€ is expensive
    (3€ is still kind of expensive, unless you're in a posh bar)

Kinds of Knowledge III

Knowledge can come in various degrees of **(un)certainty**:

- **crisp/certain** knowledge: $2 + 3 = 5$
  (we can assign a binary truth value to statements)
- **uncertain** knowledge
  - **coarse** knowledge: sun sets around 18:00 – 21:00
    (certainly, it will set some time between 18:00 and 21:00)
  - **vague** knowledge: a soft drink for 4€ is expensive
    (3€ is still kind of expensive, unless you're in a posh bar)
  - **estimated** knowledge: time from a clock reading 9:22
    (to the best of our knowledge, but clock is not exact)

Kinds of Knowledge III

Knowledge can come in various degrees of **(un)certainty**:

- **crisp/certain** knowledge: $2 + 3 = 5$
  (we can assign a binary truth value to statements)
- **uncertain** knowledge
  - **coarse** knowledge: sun sets around 18:00 – 21:00
    (certainly, it will set some time between 18:00 and 21:00)
  - **vague** knowledge: a soft drink for 4€ is expensive
    (3€ is still kind of expensive, unless you're in a posh bar)
  - **estimated** knowledge: time from a clock reading 9:22
    (to the best of our knowledge, but clock is not exact)
  - **probable** knowledge: $P(\text{sun sets after 20:00}) = 0.87$
    (NB: we're certain about the probability)

Knowledge can come in various degrees of **(un)certainty**:

- **crisp/certain** knowledge: $2 + 3 = 5$
  (we can assign a binary truth value to statements)
- **uncertain** knowledge
  - **coarse** knowledge: sun sets around 18:00 – 21:00
    (certainly, it will set some time between 18:00 and 21:00)
  - **vague** knowledge: a soft drink for 4€ is expensive
    (3€ is still kind of expensive, unless you're in a posh bar)
  - **estimated** knowledge: time from a clock reading 9:22
    (to the best of our knowledge, but clock is not exact)
  - **probable** knowledge: $P(\text{sun sets after 20:00}) = 0.87$
    (NB: we're certain about the probability)
  - **undetermined** knowledge: Germany wins Fifa cup 2022
    (will be certain in about three years' time)

Adopted from Davis, Shrobe, and Szelovits (1993):

1. **KR is a surrogate**
   imperfections are inevitable, wrong results will occur

Adopted from Davis, Shrobe, and Szelovits (1993):

1. **KR is a surrogate**
2. **KR is a set of ontological commitments**
   KR implements a particular view on the world

Adopted from Davis, Shrobe, and Szelovits (1993):

1. **KR is a surrogate**
2. **KR is a set of ontological commitments**
3. **KR is a fragmentary theory of intelligent reasoning**
   specifics of a KR decide which conclusions can be draw

Adopted from Davis, Shrobe, and Szelovits (1993):

1. **KR is a surrogate**
2. **KR is a set of ontological commitments**
3. **KR is a fragmentary theory of intelligent reasoning**
4. **KR is a medium for efficient computation**
   layout of a KR affects efficiency of algorithms

Adopted from Davis, Shrobe, and Szelovits (1993):

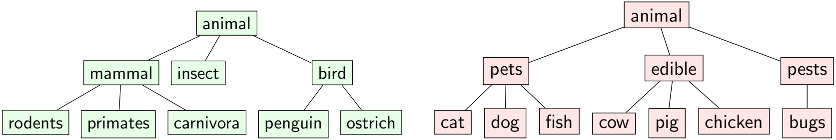
1. **KR is a surrogate**
2. **KR is a set of ontological commitments**
3. **KR is a fragmentary theory of intelligent reasoning**
4. **KR is a medium for efficient computation**
5. **KR is a medium for interaction with humans**
   Knowledge bases may be need filled in by humans, inspected, or linked to human-generated data

Adopted from Davis, Shrobe, and Szelovits (1993):

1. **KR is a surrogate**
2. **KR is a set of ontological commitments**
3. **KR is a fragmentary theory of intelligent reasoning**
4. **KR is a medium for efficient computation**
5. **KR is a medium for interaction with humans**

Roles are mutually competing! We have already discussed Role 1 and we will discuss Roles 3-5 in context of specific KRs.
$\rightsquigarrow$ We refer to these roles when evaluating utility of a KR for some purpose.

Ontology: branch of metaphysics dealing with the nature of being.
Consider two hierarchies for representing animals:



Ontological commitments by the designers vary!

⇝ Trade-offs between different views need to be considered.

KR vs. Programming

Similar to programming, both have their languages: **knowledge representation languages** and **programming languages**

- KRs may be manually instantiated similar to programs
- In programming, we design **abstract data structures**
  - choosing an internal data structures that makes certain functions very efficient, e.g., a binary tree for search
  - such data structure may be poor for other tasks, though
  - characterising the data structure by the methods offered
- In KR, we also **design representations**
  - choosing a layout of facts that enables certain inferences and allows them to be efficiently achieved
  - chosen layout may not allow for other inferences, though
  - characterising the representation by the task that can be accomplished with it

$\leadsto$ KR is similar to design of data types and algorithms, but **abstracts** from technical details of data structures.

KR vs. Programming II

- Almost all programming languages are used to denote **procedures** to manipulate data, every piece of a program has a single direction

$$\texttt{lookup-phone-number} :: \text{Person} \rightarrow \text{Number}$$

- By contrast, knowledge is **undirected**: If you know the number of a friend...
  - you can remember his or her number to dial it
  - you can recognise your friend's number when you see the number

$\rightsquigarrow$ Knowledge representation languages thus have different **semantics** as compared to programming languages

Like with programming languages, any knowledge representation requires

syntax formal language, defines how facts can be written

semantics defining their meaning

In context of programming languages, semantics are respected by compiler or interpreter to obtain a semantically equivalent piece of software the computer can execute.
In KR, semantics will be exploited to design a set of reasoning algorithms. Unlike with programming languages[1], KR researchers are not picky about syntax.

- s-expressions and formal logics are most common

[1] "I don't like my program ending))))), I prefer }}}}}!

Systems employing an architecture based on a central knowledge representation are called **knowledge-based systems**.

- Classically, knowledge bases were designed manually: **knowledge engineering**
- Today, knowledge extraction from external sources is investigated, e.g., Wikipedia

Knowledge-based systems have lost much attention due to advances in machine learning.

- However, knowledge-based systems are well-suited to **open-ended tasks**
- For example dealing with novel situations for which no training data was provided
- Knowledge-based systems are no black boxes, they can be analysed, e.g., **software verfication**

Part I: KR

In 1974, Marvin Minsky proposes **frames** to represent knowledge required for intelligent decisions:

- Idea is representing a prototypical situation, e.g., "visiting a restaurant"
- Frames may comprise **sub-frames**, e.g., "visiting rock concert" may have sub-frame "visiting bar"
- Frame information is organised in **slots**, e.g., "visiting a restaurant" would define a slot for entrance door, waiter, etc.
- **Default values** may be provided for slots
- **Procedural attachments** may be provided, e.g., "tipping the waiter"

Frames have not been defined formally, assessing their relevance is thus difficult.
Observe that the idea of frames lives on in today's **object oriented programming (OOP)**.

- Slots, default values and inheritance
- Procedural attachments are called methods

$\rightsquigarrow$ Frames include several advanced mechanisms not part of mainstream OOP languages! Compare frames or Lisp's CLOS/MOP to Java...

$\rightsquigarrow$ OOP has its roots in AI knowledge representation!

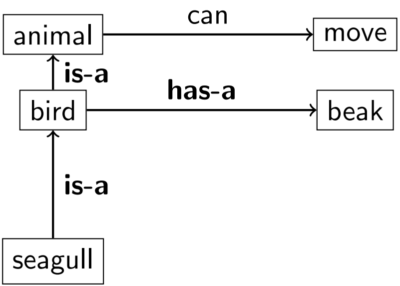
Semantic Networks are considered older than AI as they have already been used as semi-formal representation to capture information in natural language texts.

- Still in use today, though formalisation and techniques have evolved
- Related areas: **formal ontology**, **semantic web**, **linked data**, ...

⇝ Knowledge graphs underlying Wikidata and **Question-Answering Systems**

Basic idea:

- Representations are composed of **entities** and **relations**
- KR is given as **labeled directed graph**



representation using s-expressions:

```
(define *kb* '((gull is-a bird)
               (bird is-a animal)
               (bird has-a beak)
                ...))
```

In particular, relations **is-a** and **has-a** are typical for all semantic networks.

Question: How can we reason with semantic networks, e.g., to conclude that seagulls have beaks, too?

In order to design reasoning procedures and analyse them, a formal semantics of relations must be provided.

Example:

$$\text{is-a}(X, Y) \wedge \text{has-a}(Y, Z) \Rightarrow \text{has-a}(X, Z)$$

In order to design reasoning procedures and analyse them, a formal semantics of relations must be provided.

Example:

$$\text{is-a}(X, Y) \wedge \text{has-a}(Y, Z) \Rightarrow \text{has-a}(X, Z)$$

What about $\text{during}(A, B) \wedge \text{overlaps}(B, C) \Rightarrow ???(A, C)$?

In order to design reasoning procedures and analyse them, a formal semantics of relations must be provided.

Example:

$$\text{is-a}(X, Y) \land \text{has-a}(Y, Z) \Rightarrow \text{has-a}(X, Z)$$

What about $\text{during}(A, B) \land \text{overlaps}(B, C) \Rightarrow ???(A, C)$?

Knowledge representations need to define semantics or reasoning procedures (e.g., rules like above) to obtain methods that draw valid conclusions.

- Motivation for **logic**-based approach to KR
- Logics comprise **syntax**, **semantics**, and **inference rules**

Proposed by Brachman and Schmolze (1985), the KL-ONE system provides an implemented language which contains many elements later found in the area of **description logics**.

- object-centred approach: **concepts** as principal element
- relations defined by **roles**
- structure-forming constructs: specialisation, restriction, etc.

KL-ONE example:

```
(cdef GARDENER (and PERSON (c-some Hobby GARDENING-ACTIVITY)))
```

In modern description logic notation:

$$\text{Gardener} \sqsubseteq \text{Person} \sqcap \exists.\text{Hobby}.\text{GardeningActivity}$$

KL-ONE Constructs

Excerpt from KL-ONE syntax Wood & Schmolze (1992):

```
<concept>        ::= top |
                     <concept-name> |
                     (and <concept>+) |
                     (or <concept>+) |
                     (not <concept>) |
                     (all <role> <concept>) |
                     (some <role>) |
                     (c-some <role> <concept>) |
                     (atleast <minimum> <role>) |
                     (c-atleast <minimum> <role> <concept>) |
                     (atmost <maximum> <role>) |
                     (c-atmost <maximum> <role> <concept>) |
                     (rvm <role> <role>) |
                     (rvm= <role> <role>) |
                     (sd <concept> (<role> <role>)+)
```

In KRs, one typically distinguishes **instances** from **concepts**:

- Fred is an instance of the concept ostrich.

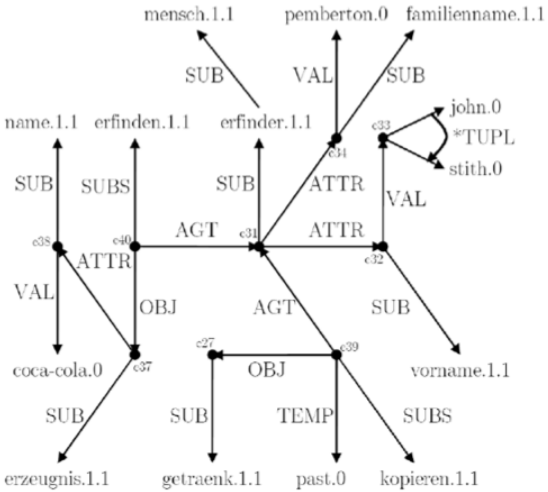
In description logics, the terms **ABox (assertion box)** and **TBox (terminology box)** have been coined.

- ABoxes contain assertions about named individuals, e.g., Fred
- TBoxes provides terminology by introducing concept names for complex descriptions

$\rightsquigarrow$ More on this in the final KR lecture.

Example

Semantic network excerpt automatically extracted from Wikipedia text, used as knowledge base for open-domain question answering system LogAnswer (Fuhrbach et al., 2010):



Aside symbolical representations, quantitative approaches exist.
Example: Word embeddings

- Idea: map every natural word w to a point $e(w) \in \mathbb{R}^N$ with N chosen in range 500–1000
- Choose mapping such that distances between words $d(e(w_1), e(w_2))$ reflect 'distances' of how they occur in natural language text
- Observation: $d(e(\text{'man'}), e(\text{'woman'})) \sim d(e(\text{'king'}), e(\text{'queen'}))$
- Thus, some of the semantics of language is retained

Current research topic: Investigate how embeddings can be designed such that geometric operations perform reasoning tasks.

Part II: Reasoning

So far, we have considered **search** as a method to perform reasoning. It can be applied in several reasoning tasks. Search in conjunction with an adequate knowledge representation allows us to tackle all standard AI tasks.

area of reasoning	**approach**
planning	search sequence of actions
configuring	search set of assignments
drawing conclusions	deciding validity by searching for a proof
learning, inductive inference	search for a set of facts/rules, from which observations follow

Interesting for a wide range of tasks, in particular:

- Satisfiability problems with symbolic and quantitative domains, e.g., entities represented in a semantic network $\rightsquigarrow$ Problem is equivalent to drawing conclusions!
- Configuration problems
- Retrieving information from a knowledge base

Constraint-based reasoning problems consist of:

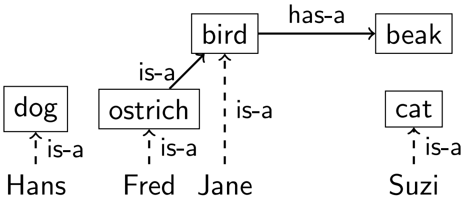
- **variables**, which represent values in an associated **domain**
- a formal language to denote constraints, the **constraint language**

Definition: A constraint satisfaction problem (CSP) is a tuple $\langle X, \mathrm{dom}, C\rangle$:

- $X = \{x_1, x_2, \ldots, x_n\}$ is a set of **variable**
- $D = \{D_1, D_2, \ldots, D_n\}$ is a set of **domains**
- $\mathrm{dom} : X \to D$ is called the **domain mapping**
- We call ϕ with $x_i \to \mathrm{dom}(x_i)$ is called **valuation**
- C is a set of **constraints**, symbolic expressions in a constraint language that involve variables from X.
- A constraint $c \in C$ is called **satisfied** (by ϕ), if, when replacing all free occurrences of $x \in X$ in c by $\phi(x)$, c evaluates to true in the given constraint language.
- A valuation ϕ satisfying all $c \in C$ is called a **solution**
- The CSP is the problem of computing such a solution

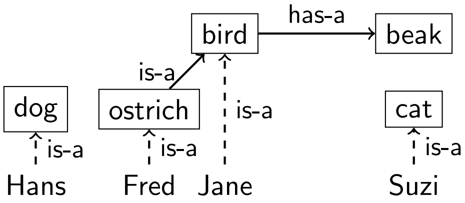
Consider a constraint language that only allows for expressions $r(X, Y)$ where r is a role in a semantic network and X, Y are either variables or constants. Let the following CSP instance $\langle X, \mathrm{dom}, C\rangle$ be given:

- $X = \{A, B\}$
- $D_1 = \{\text{Hans}, \text{Jane}, \text{Fred}, \text{Suzi}, \text{ostrich}, \text{bird}, \text{beak}, \text{dog}, \text{cat}\}$, $\mathrm{dom}(A) = \mathrm{dom}(B) = D_1$
- $C = \{\text{is-a}(A, B), \text{has-a}(A, \text{beak})\}$



Consider a constraint language that only allows for expressions $r(X, Y)$ where r is a role in a semantic network and X, Y are either variables or constants. Let the following CSP instance $\langle X, \mathrm{dom}, C \rangle$ be given:

- $X = \{A, B\}$
- $D_1 = \{\text{Hans}, \text{Jane}, \text{Fred}, \text{Suzi}, \text{ostrich}, \text{bird}, \text{beak}, \text{dog}, \text{cat}\}$, $\mathrm{dom}(A) = \mathrm{dom}(B) = D_1$
- $C = \{\text{is-a}(A, B), \text{has-a}(A, \text{beak})\}$



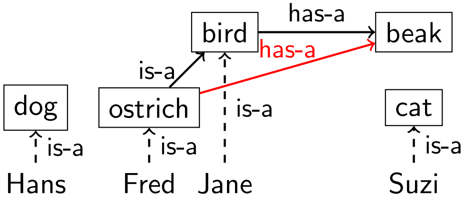
$\phi(A) = \text{Jane}$
$\phi(B) = \text{bird}$

Example CSP

Consider a constraint language that only allows for expressions $r(X, Y)$ where r is a role in a semantic network and X, Y are either variables or constants. Let the following CSP instance $\langle X, \mathrm{dom}, C\rangle$ be given:

- $X = \{A, B\}$
- $D_1 = \{\text{Hans}, \text{Jane}, \text{Fred}, \text{Suzi}, \text{ostrich}, \text{bird}, \text{beak}, \text{dog}, \text{cat}\}$, $\mathrm{dom}(A) = \mathrm{dom}(B) = D_1$
- $C = \{\text{is-a}(A, B), \text{has-a}(A, \text{beak})\}$

$\text{is-a}(X, Y) \wedge \text{has-a}(Y, Z) \Rightarrow \text{has-a}(X, Z)$

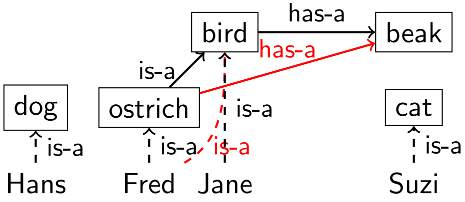


$\phi(A) = \text{Fred}$
$\phi(B) = \text{ostrich}$

Consider a constraint language that only allows for expressions $r(X, Y)$ where r is a role in a semantic network and X, Y are either variables or constants. Let the following CSP instance $\langle X, \mathrm{dom}, C\rangle$ be given:

- $X = \{A, B\}$
- $D_1 = \{\text{Hans}, \text{Jane}, \text{Fred}, \text{Suzi}, \text{ostrich}, \text{bird}, \text{beak}, \text{dog}, \text{cat}\}$, $\mathrm{dom}(A) = \mathrm{dom}(B) = D_1$
- $C = \{\text{is-a}(A, B), \text{has-a}(A, \text{beak})\}$

$$\text{is-a}(X, Y) \wedge \text{is-a}(Y, Z) \Rightarrow \text{is-a}(X, Z)$$



$\phi(A) = \text{Fred}$
$\phi(B) = \text{bird}$

Solving CSPs differs in complexity, depending on the respective constraint language.

- solving CSPs goes far beyond querying a knowledge base as shown in example
- even for simple constraint languages, complexity is NP-complete for finite domains
- CSPs allow us to model a great variety of problems, including solving a Sudoku, configuring industrial processes, verifying software, ...

Theorem

Computing a solution for a CSP is NP-complete, even for constraint languages that allow satisfiability to be checked in $O(n)$ time, where n is the length the CSP encoding.

Proof Sketch.

Consider **circuit algebra** as constraint language:

- terms comprising Boolean variables x_i and operators $\neg, \vee, \wedge$
- constraints are satisfied if the evaluate to 'true'
- terms can be evaluated in $O(n)$ time

NP containment: guess a solution $X \rightarrow \{\text{true, false}\}$ and verify it.
NP hardness by polynomial-time many-one reduction from Boolean satisfiability (SAT, known to be NP-complete) to CSP: Let a Boolean formula in CNF be given

$$\underbrace{(\neg x_1 \vee x_2 \vee \neg x_3)}_{C_1} \wedge \underbrace{(x_2 \vee \neg x_4)}_{C_2} \wedge \ldots$$

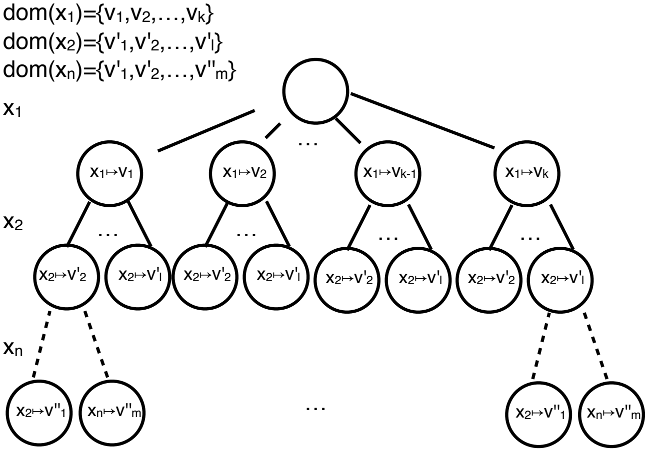
Set $C = \{C_1, C_2, \ldots C_k\}$ we the set of clauses. Every solution to this CSP instance is a solution to the given SAT instance. □

Sophisticated algorithms have been developed to solve various kinds of CSPs.

$\rightsquigarrow$ more in Master's course on KR

General principle is easy: **search** As state space consider the **association tree**: On level l, the lth variable is assigned to a value. Search for a leaf on level $|X|$ that satisfies all constraints.

- Useful pruning technique to make search efficient presented in tutorials



$\prod_{i=1}^{n} |\mathrm{dom}(x_i)|$ leafs

Summary

- The field of **Knowledge Representation (KR)** and **reasoning**
- **Knowledge** vs. **information**
- Various kinds of knowledge – gives rise to plenty KR approaches
- Five mutually competing roles: surrogate, ontological commitments, fragmentary theory, medium for efficient communication, medium for interaction
- Fundamental KR approaches: **frames**, **semantics networks**
- reasoning with **constraints**: **Constraint Satisfaction Problem (CSP)**

- S. Russel & P. Norvig (2010, 3rd edition). Artificial Intelligence: A Modern Approach, Chapter 6 "Agents that reason logically", Chapter 8 "Building a Knowledge Base"
- R. J. Brachman & H. J. Levesque (2004). Knowledge Representation and Reasoning, Elsevier, available online
- R. Davis, H. Shrobe, & P. Szolovits (1993). What is a Knowledge Representation, AI Magazine, 14(1):17–33
- R. J. Brachman & J. G. Schmolze (1985). An Overview of the KL-ONE Knowledge Representation System, Cognitive Science, 9(2): 171-216
- W. A. Woods & J. G. Schmolze (1992). The KL-ONE Family, Computers and Mathematics with Applications, 23(2–5):133–177
- U. Furbach, I. Glöckner, & B. Pelzer (2010). An application of automated reasoning in natural language question answering. AI Commun. 23(2–3): 241–265