

Sémantické sítě a rámce

Petr Křemen

Katedra kybernetiky, FEL ČVUT

Co nás čeká

- 1 Úvod do reprezentace znalostí
- 2 Sémantické sítě
- 3 Rámce

Motivace

Proč se budeme zabývat reprezentací znalostí na obecné bázi ?

- **lidé** potřebují rychle a efektivně vyhledávat relevantní informace. Je mnoho způsobů, jak reprezentovat danou doménu – např. na léku bude něco jiného zajímat lékaře, něco jiného lékárníky, apod.

Motivace

Proč se budeme zabývat reprezentací znalostí na obecné bázi ?

- **lidé** potřebují rychle a efektivně vyhledávat relevantní informace. Je mnoho způsobů, jak reprezentovat danou doménu – např. na léku bude něco jiného zajímat lékaře, něco jiného lékárníky, apod.
- obtížné je vytvořit reprezentaci dané znalosti využitelnou pro více úloh.

Motivace

Proč se budeme zabývat reprezentací znalostí na obecné bázi ?

- **lidé** potřebují rychle a efektivně vyhledávat relevantní informace. Je mnoho způsobů, jak reprezentovat danou doménu – např. na léku bude něco jiného zajímat lékaře, něco jiného lékárníky, apod.
- obtížné je vytvořit reprezentaci dané znalosti využitelnou pro více úloh.
- efektivní reprezentace znalostí představuje základ pro **strojové** zpracování dat – inferenční mechanismy.

Motivace (2)

My se budeme zabývat možnostmi deklarativní reprezentace znalostí:

- databáze → vyhledávání (*K čemu slouží indexy v relačních databázích ?, K čemu pohledy (views) ?*)

Motivace (2)

My se budeme zabývat možnostmi deklarativní reprezentace znalostí:

- databáze → vyhledávání (*K čemu slouží indexy v relačních databázích ?, K čemu pohledy (views) ?*)
- sémantické servery, sémantický web → vyhledávání

Motivace (2)

My se budeme zabývat možnostmi deklarativní reprezentace znalostí:

- databáze → vyhledávání (*K čemu slouží indexy v relačních databázích ?, K čemu pohledy (views) ?*)
- sémantické servery, sémantický web → vyhledávání
- multiagentní technologie → komunikační obsah

Motivace (2)

My se budeme zabývat možnostmi deklarativní reprezentace znalostí:

- databáze → vyhledávání (*K čemu slouží indexy v relačních databázích ?, K čemu pohledy (views) ?*)
- sémantické servery, sémantický web → vyhledávání
- multiagentní technologie → komunikační obsah
- strojové učení → jazykový bias

Motivace (2)

My se budeme zabývat možnostmi deklarativní reprezentace znalostí:

- databáze → vyhledávání (*K čemu slouží indexy v relačních databázích ?, K čemu pohledy (views) ?*)
- sémantické servery, sémantický web → vyhledávání
- multiagentní technologie → komunikační obsah
- strojové učení → jazykový bias
- **Volba vhodné reprezentace znalostí je jednou z nejdůležitějších otázek při strojovém řešení téměř všech problémů AI.**

Motivace (2)

My se budeme zabývat možnostmi deklarativní reprezentace znalostí:

- databáze → vyhledávání (*K čemu slouží indexy v relačních databázích ?, K čemu pohledy (views) ?*)
- sémantické servery, sémantický web → vyhledávání
- multiagentní technologie → komunikační obsah
- strojové učení → jazykový bias
- **Volba vhodné reprezentace znalostí je jednou z nejdůležitějších otázek při strojovém řešení téměř všech problémů AI.**
- Nyní se budeme zabývat *reprezentací znalostí bez neurčitosti.*

Jak reprezentovat znalosti v umělé inteligenci.

Vše začalo snahou lidí strukturovat si své znalosti a snažit se formalizovat lidské uvažování.

- Logika, Prolog

Jak reprezentovat znalosti v umělé inteligenci.

Vše začalo snahou lidí strukturovat si své znalosti a snažit se formalizovat lidské uvažování.

- Logika, Prolog
- Sémantické sítě (Semantic Networks) - Quillian, 1967.

Jak reprezentovat znalosti v umělé inteligenci.

Vše začalo snahou lidí strukturovat si své znalosti a snažit se formalizovat lidské uvažování.

- Logika, Prolog
- Sémantické sítě (Semantic Networks) - Quillian, 1967.
- Rámce (Frames) - Minsky, 1974.

Jak reprezentovat znalosti v umělé inteligenci.

Vše začalo snahou lidí strukturovat si své znalosti a snažit se formalizovat lidské uvažování.

- Logika, Prolog
- Sémantické sítě (Semantic Networks) - Quillian, 1967.
- Rámce (Frames) - Minsky, 1974.
- Relační databáze

Jak reprezentovat znalosti v umělé inteligenci.

Vše začalo snahou lidí strukturovat si své znalosti a snažit se formalizovat lidské uvažování.

- Logika, Prolog
- Sémantické sítě (Semantic Networks) - Quillian, 1967.
- Rámce (Frames) - Minsky, 1974.
- Relační databáze
- Objektově-orientované programovací jazyky

Jak reprezentovat znalosti v umělé inteligenci.

Vše začalo snahou lidí strukturovat si své znalosti a snažit se formalizovat lidské uvažování.

- Logika, Prolog
- Sémantické sítě (Semantic Networks) - Quillian, 1967.
- Rámce (Frames) - Minsky, 1974.
- Relační databáze
- Objektově-orientované programovací jazyky
- Deskripční logiky a sémantický web (RDF, RDFS, OIL, DAML+OIL, OWL)

Produkční systémy

Báze dat – obsahuje “model světa”

Produkční systémy

Báze dat – obsahuje “model světa”

Produkční pravidla – pravidla ve tvaru

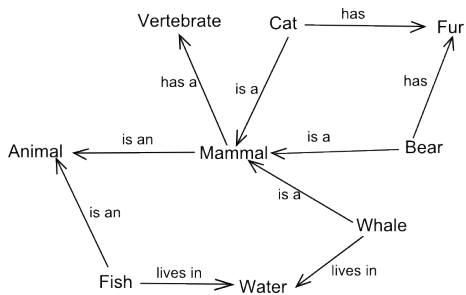
Produkční systémy

Báze dat – obsahuje “model světa”

Produkční pravidla – pravidla ve tvaru

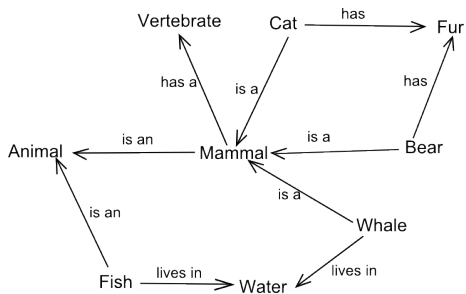
Inferenční stroj – Vybírá (jak ?) pravidla, provádí akce, ...

Sémantické síťě



(©wikipedia.org)

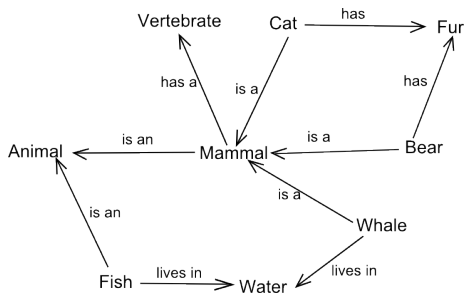
Sémantické sítě



(©wikipedia.org)

- Uzly jsou tvořeny entitami (instance, třídy), hrany reprezentují binární relace.

Sémantické sítě



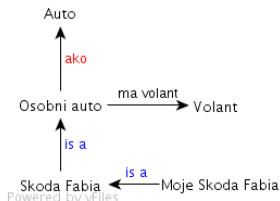
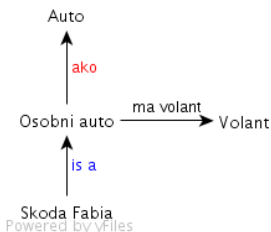
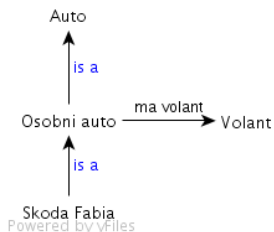
(©wikipedia.org)

- Uzly jsou tvořeny entitami (instance, třídy), hrany reprezentují binární relace.
- Jediná inference je dědění pomocí **is a** relace.

Příklad

Each Cat **has a** Vertebrate, since each Cat **is a** Mammal.

Sémantické sítě (2)



Tento způsob však nerozlišuje instance a třídy a odpovídá tak *logice vyššího řádu*.

Řešením je zavést nový typ relace **ako** a používat ji pro dědičnost, zatímco **is a** používat pro instanciování.

Ani zavedení relace **ako** neřeší zcela tento problém.

Sémantické sítě (3)

- 😊 jsou velmi jednoduché - z logického pohledu se jedná o jednoduchou binární relační strukturu – vyjma relací **has a** a **is a**. Sémantiku těchto relací lze vyjádřit pravidlem
 $has_a(?x, ?y), is_a(?z, ?x) \rightarrow has_a(?z, ?y)..$

Sémantické sítě (3)

- 😊 jsou velmi jednoduché - z logického pohledu se jedná o jednoduchou binární relační strukturu – vyjma relací **has a** a **is a**. Sémantiku těchto relací lze vyjádřit pravidlem
 $has_a(?x, ?y), is_a(?z, ?x) \rightarrow has_a(?z, ?y)..$
- ☹️ neumožňují vyjádřit nemonotónní znalosti (podobně jako FOL).

Sémantické sítě (3)

- 😊 jsou velmi jednoduché - z logického pohledu se jedná o jednoduchou binární relační strukturu – vyjma relací **has a** a **is a**. Sémantiku těchto relací lze vyjádřit pravidlem
 $has_a(?x, ?y), is_a(?z, ?x) \rightarrow has_a(?z, ?y)..$
- ☹️ neumožňují vyjádřit nemonotónní znalosti (podobně jako FOL).
- ☹️ neumožňují vyjádřit n-ární relace. Ty je nutné nejprve **reifikovat**.

Sémantické sítě (3)

- 😊 jsou velmi jednoduché - z logického pohledu se jedná o jednoduchou binární relační strukturu – vyjma relací **has a** a **is a**. Sémantiku těchto relací lze vyjádřit pravidlem $has_a(?x, ?y), is_a(?z, ?x) \rightarrow has_a(?z, ?y)..$
- ☹️ neumožňují vyjádřit nemonotónní znalosti (podobně jako FOL).
- ☹️ neumožňují vyjádřit n-ární relace. Ty je nutné nejprve **reifikovat**.
- ☹️ neumožňují vyjádřit složitější konstrukty, jako kardinality: “Každý člověk má nejvýše dvě nohy.”, dědení vlastností “každý otec je i rodičem”, aj.

Sémantické sítě (3)

- ☺ jsou velmi jednoduché - z logického pohledu se jedná o jednoduchou binární relační strukturu – vyjma relací **has a** a **is a**. Sémantiku těchto relací lze vyjádřit pravidlem $has_a(?x, ?y), is_a(?z, ?x) \rightarrow has_a(?z, ?y)..$
- ☹ neumožňují vyjádřit nemonotónní znalosti (podobně jako FOL).
- ☹ neumožňují vyjádřit n-ární relace. Ty je nutné nejprve **reifikovat**.
- ☹ neumožňují vyjádřit složitější konstrukty, jako kardinality: “Každý člověk má nejvýše dvě nohy.”, dědení vlastností “každý otec je i rodičem”, aj.
- Wordnet, sémantické wiki, aj.

Sémantické sítě – Wordnet

hyponyma, hypernyma odpovídají **is-a** relaci (příp. **ako** relaci).

Sémantické sítě – Wordnet

hyponyma, hypernyma odpovídají **is-a** relaci (příp. **ako** relaci).
meronyma, holonyma odpovídají **has-a** relaci

Sémantické sítě – Wordnet

hyponyma, hypernyma odpovídají **is-a** relaci (příp. **ako** relaci).

meronyma, holonyma odpovídají **has-a** relaci

synonyma, antonyma ?

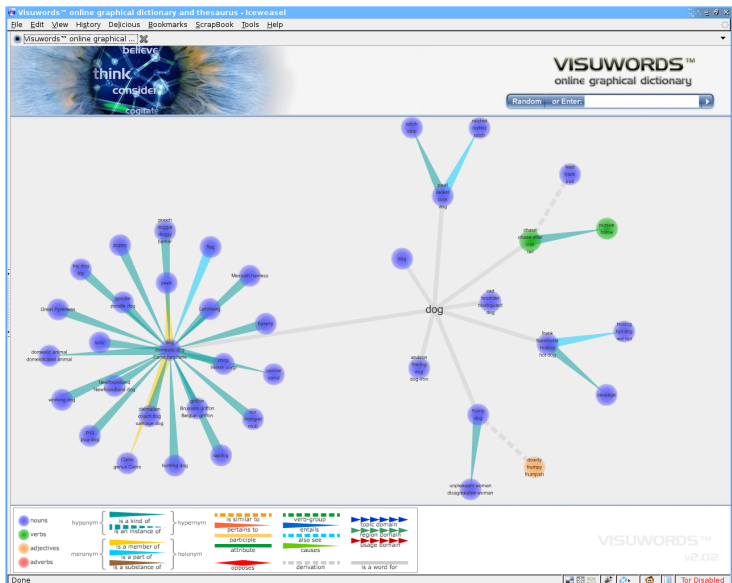
Sémantické sítě – Wordnet

hyponyma, hypernyma odpovídají **is-a** relaci (příp. **ako** relaci).

meronyma, holonyma odpovídají **has-a** relaci

synonyma, antonyma ?

<http://wordnet.princeton.edu/>

Sémantické sítě – <http://www.visuwords.com/>

Rámce

frame: Škoda Favorit

slots:

is a: osobní auto

ma motor: čtyřdobý benzínový

ma převodovku: manuální

ma karburator: *hodnota:* Jikov

předpoklad: Pierburg

- strukturovanější než SN

([MvL93])

Rámce

frame: Škoda Favorit

slots:

is a: osobní auto

ma motor: čtyřdobý benzínový

ma převodovku: manuální

ma karburator: *hodnota:* Jikov

předpoklad: Pierburg

- strukturovanější než SN
- formuláře, které obsahují **sloty** (binární relace).

([MvL93])

Rámce

frame: Škoda Favorit

slots:

is a: osobní auto

ma motor: čtyřdobý benzínový

ma převodovku: manuální

ma karburator: *hodnota:* Jikov
předpoklad: Pierburg

- Každý slot může mít několik **facetů** (omezení na používání slotu), např. kardinalitu, defaultní hodnotu, apod.

- strukturovanější než SN
- formuláře, které obsahují **sloty** (binární relace).

([MvL93])

Rámce

frame: Škoda Favorit

slots:

is a: osobní auto

ma motor: čtyřdobý benzínový

ma převodovku: manuální

ma karburator: *hodnota:* Jikov
předpoklad: Pierburg

- Každý slot může mít několik **facetů** (omezení na používání slotu), např. kardinalitu, defaultní hodnotu, apod.
- ☺ Facety nám umožňují nemonotónní odvozování.

- strukturovanější než SN
- formuláře, které obsahují **sloty** (binární relace).

([MvL93])

Rámce

frame: Škoda Favorit

slots:

is a: osobní auto

ma motor: čtyřdobý benzínový

ma převodovku: manuální

ma karburator: *hodnota:* Jikov
předpoklad: Pierburg

C

- strukturovanější než SN
- formuláře, které obsahují **sloty** (binární relace).

([MvL93])

- Každý slot může mít několik **facetů** (omezení na používání slotu), např. kardinalitu, defaultní hodnotu, apod.
- ☺ Facety nám umožňují nemonotónní odvozování.
- ☺ Lze definovat *démony* – triggerery pro akce prováděné na facetech (čtení, změna, smazání). Lze je použít např. pro ověřování konzistence.

Rámce (2)

Příklad

Škoda Favorit typicky **má karburátor** typu Pierburg, tento Škoda Favorit však **má karburátor** typu Jikov.

- rámce lze sdružovat do *scénářů*. Ty představují typické situace, např. návštěva restaurace, apod. [MvL93]

Rámce (2)

Příklad

Škoda Favorit typicky **má karburátor** typu Pierburg, tento Škoda Favorit však **má karburátor** typu Jikov.

- rámce lze sdružovat do *scénářů*. Ty představují typické situace, např. návštěva restaurace, apod. [MvL93]
- OKBC - <http://www.ai.sri.com/okbc>

Rámce (2)

Příklad

Škoda Favorit typicky **má karburátor** typu Pierburg, tento Škoda Favorit však **má karburátor** typu Jikov.

- rámce lze sdružovat do *scénářů*. Ty představují typické situace, např. návštěva restaurace, apod. [MvL93]
- OKBC - <http://www.ai.sri.com/okbc>
- Protégé - <http://protege.stanford.edu/overview/protege-frames.html>

Rámce (2)

Příklad

Škoda Favorit typicky **má karburátor** typu Pierburg, tento Škoda Favorit však **má karburátor** typu Jikov.

- rámce lze sdružovat do *scénářů*. Ty představují typické situace, např. návštěva restaurace, apod. [MvL93]
- OKBC - <http://www.ai.sri.com/okbc>
- Protégé - <http://protege.stanford.edu/overview/protege-frames.html>
- Apollo - <http://apollo.open.ac.uk>

Rámce (2)

Příklad

Škoda Favorit typicky **má karburátor** typu Pierburg, tento Škoda Favorit však **má karburátor** typu Jikov.

- rámce lze sdružovat do *scénářů*. Ty představují typické situace, např. návštěva restaurace, apod. [MvL93]
- OKBC - <http://www.ai.sri.com/okbc>
- Protégé - <http://protege.stanford.edu/overview/protege-frames.html>
- Apollo - <http://apollo.open.ac.uk>
- Apollo CH - <http://labe.felk.cvut.cz/falc/Apollo>

Rámce (3) - Apollo CH

The screenshot shows the Apollo 0.28.0 application window. The interface is divided into several panes:

- Left Pane:** A hierarchical tree of knowledge objects. The root is 'default', which contains 'time-and-date' and 'diagnose-and-fix'. 'diagnose-and-fix' contains 'general-medical-knowledge'. The 'general-medical-knowledge' pane is expanded, showing a list of classes: 'drug', 'my-class', 'blood-pressure', 'disease-disorder' (highlighted), 'cerebrovascular-disease', 'heart-disease', 'renal-disease', 'vascular-disease', 'medical-condition-disorder', 'ace-inhibitors', 'advanced-hypertensive-retinopathy', 'alpha-blockers', 'angiotensin-2-antagonists', 'beta-blockers', 'calcium-antagonists', and 'cerebrovascular-disease'. The status bar at the bottom indicates 'Current : general-medical-knowledge'.
- Top-Right Panes:**
 - Super-classes:** Lists 'disorder'.
 - Slots using focus:** Currently empty.
- Center Pane:** A detailed view of the 'disease-disorder' class. It features a table with the following columns: Slot, Type, Value, Cardinality, and Documen.

Slot	Type	Value	Cardinality	Documen.
description	string		R	
- Bottom-Right Panes:**
 - Documentation:** Contains tabs for 'Documentation', 'Slots', and 'Relational'.
 - Sub-classes and instances:** Lists 'cerebrovascular-disease', 'heart-disease', 'renal-disease', and 'vascular-disease'.
 - Classes used by focus:** Lists 'string'.

Rámce (4) - Protégé

newspaper Protégé 3.2.1 (file:/home/kremen/programs/Protege_3.2.1/examples/newspaper/newspaper.pprj, Protégé Files (.pont and .pins))

File Edit Project Window Tools Help

Classes Slots Forms Instances Queries

CLASS BROWSER

For Project: newspaper

Class Hierarchy

- THING
 - SYSTEM-CLASS
 - META-CLASS
 - CLASS
 - STANDARD-CLASS
 - SLOT
 - STANDARD-SLOT

- Author
- News_Service
- Columnist
- Editor
- Reporter
- Content
 - Advertisement
 - Personals_Ad
 - Standard_Ad
 - Article
 - Library
 - Newspaper

CLASS EDITOR

For Class: :STANDARD-SLOT (instance of :STANDARD-CLASS)

Name: :STANDARD-SLOT

Documentation:

Constraints:

Role: Concrete

Template Slots

Name	Cardinality	Type	Other Facets
:ASSOCIATED-FACET	single	Instance of FACET	inverse-slot=:ASSOCIATED-SLOT
:DIRECT-DOMAIN	multiple	Instance of CLASS	inverse-slot=:DIRECT-TEMPLATE-SLOTS
:DIRECT-SUBSLOTS	multiple	Instance of SLOT	inverse-slot=:DIRECT-SUBSLOTS
:DIRECT-SUPERSLOTS	multiple	Instance of SLOT	inverse-slot=:DIRECT-SUBSLOTS
:DIRECT-TYPE	multiple	Class with superclass :SLOT	inverse-slot=:DIRECT-INSTANCES
:DOCUMENTATION	multiple	String	
:NAME	single	String	
:SLOT-CONSTRAINTS	multiple	Instance of :CONSTRAINT	
:SLOT-DEFAULTS	multiple	Any	
:SLOT-INVERSE	single	Instance of :SLOT	inverse-slot=:SLOT-INVERSE
:SLOT-MAXIMUM-CARDINALITY	single	Integer	default=1
:SLOT-MINIMUM-CARDINALITY	single	Integer	
:SLOT-NUMERIC-MAXIMUM	single	Float	
:SLOT-NUMERIC-MINIMUM	single	Float	
:SLOT-VALUE-TYPE	multiple	Any	default=String
:SLOT-VALUES	multiple	Any	

Superclasses

- SLOT

Rámce a sémantické sítě - shrnutí

- 😊 velmi jednoduché struktury pro reprezentaci znalostí,

Rámce a sémantické sítě - shrnutí

- 😊 velmi jednoduché struktury pro reprezentaci znalostí,
- 😊 nemonotónní odvozování,

Rámce a sémantické sítě - shrnutí

- 😊 velmi jednoduché struktury pro reprezentaci znalostí,
- 😊 nemonotónní odvozování,
- 😞 ad-hoc odvozovací procedury, překlad do FOL není jednoduchý, a tedy ani jednoznačný,

Rámce a sémantické sítě - shrnutí

- 😊 velmi jednoduché struktury pro reprezentaci znalostí,
- 😊 nemonotónní odvozování,
- 😞 ad-hoc odvozovací procedury, překlad do FOL není jednoduchý, a tedy ani jednoznačný,
- 😞 problémy – dotazovací jazyk, debugging.