

Konceptuální grafy

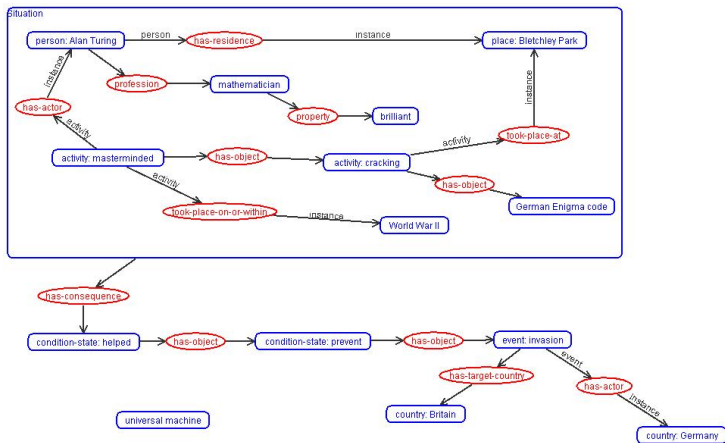
Petr Křemen

Katedra kybernetiky, FEL ČVUT

Co nás čeká

1 Konceptuální grafy

Příklad



Konceptuální grafy

konceptuální graf je bipartitní graf s uzly typu (1) **koncept** a (2) **relace**.

Konceptuální grafy

konceptuální graf je bipartitní graf s uzly typu (1) **koncept** a (2) **relace**.
koncept je tvaru **typ-konceptu : referent**.

Konceptuální grafy

konceptuální graf je bipartitní graf s uzly typu (1) **koncept** a (2) **relace**.
koncept je tvaru **typ-konceptu : referent**.

Příklad (Typy kvantifikátorů)

dog : Lucky

“Pes Lucky”

$\exists x \text{Pes}(x) \wedge \text{Name}(x, \text{Lucky}) \wedge \dots$

dog

“Nějaký pes”

$\exists x \text{Pes}(x) \wedge \dots$

dog : \forall

“Všichni psi”

$\forall x \text{Pes}(x) \rightarrow \dots$

dog : {*}

“Množina psů”

☹ není FOL

Konceptuální grafy

konceptuální graf je bipartitní graf s uzly typu (1) **koncept** a (2) **relace**.
 koncept je tvaru **typ-konceptu : referent**.

Příklad (Typy kvantifikátorů)

dog : Lucky

“Pes Lucky”

$\exists x \text{Pes}(x) \wedge \text{Name}(x, \text{Lucky}) \wedge \dots$

dog

“Nějaký pes”

$\exists x \text{Pes}(x) \wedge \dots$

dog : \forall

“Všichni psi”

$\forall x \text{Pes}(x) \rightarrow \dots$

dog : {*}

“Množina psů”

☹ není FOL

konceptuální relace = vztah = predikát libovolné arity > 0 .

Konceptuální grafy

konceptuální graf je bipartitní graf s uzly typu (1) **koncept** a (2) **relace**.
koncept je tvaru **typ-konceptu : referent**.

Příklad (Typy kvantifikátorů)

dog : Lucky

“Pes Lucky”

$\exists x \text{Pes}(x) \wedge \text{Name}(x, \text{Lucky}) \wedge \dots$

dog

“Nějaký pes”

$\exists x \text{Pes}(x) \wedge \dots$

dog : \forall

“Všichni psi”

$\forall x \text{Pes}(x) \rightarrow \dots$

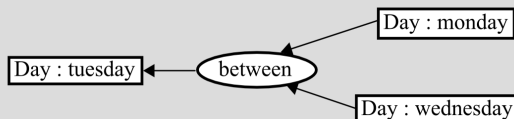
dog : {*}

“Množina psů”

☹ není FOL

konceptuální relace = vztah = predikát libovolné arity > 0 .

Příklad (ternární relace)



Konceptuální grafy (2)

referent se skládá (viz. předchozí slide) z **kvantifikátoru** (existenciální, nebo definovaný (univerzální, kolektivní, aj.)), **designátoru** (ten označuje identifikátor instance, např. jméno) a případně tzv. **deskriptoru** (konceptuální graf popisující daný koncept).

Konceptuální grafy (2)

referent se skládá (viz. předchozí slide) z **kvantifikátoru** (existenciální, nebo definovaný (univerzální, kolektivní, aj.)), **designátoru** (ten označuje identifikátor instance, např. jméno) a případně tzv. **deskriptoru** (konceptuální graf popisující daný koncept).

kontext je koncept s neprázdným deskriptorem

Konceptuální grafy (2)

referent se skládá (viz. předchozí slide) z **kvantifikátoru** (existenciální, nebo definovaný (univerzální, kolektivní, aj.)), **designátoru** (ten označuje identifikátor instance, např. jméno) a případně tzv. **deskriptoru** (konceptuální graf popisující daný koncept).

kontext je koncept s neprázdným deskriptorem

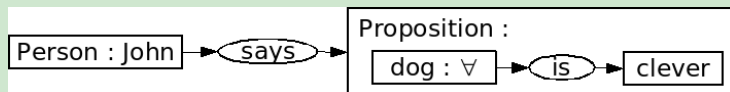
Příklad (Kontext)

Konceptuální grafy (2)

referent se skládá (viz. předchozí slide) z **kvantifikátoru** (existenciální, nebo definovaný (univerzální, kolektivní, aj.)), **designátoru** (ten označuje identifikátor instance, např. jméno) a případně tzv. **deskriptoru** (konceptuální graf popisující daný koncept).

kontext je koncept s neprázdným deskriptorem

Příklad (Kontext)

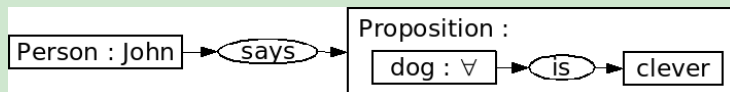


Konceptuální grafy (2)

referent se skládá (viz. předchozí slide) z **kvantifikátoru** (existenciální, nebo definovaný (univerzální, kolektivní, aj.)), **designátoru** (ten označuje identifikátor instance, např. jméno) a případně tzv. **deskriptoru** (konceptuální graf popisující daný koncept).

kontext je koncept s neprázdným deskriptorem

Příklad (Kontext)



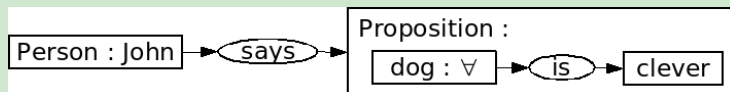
“John říká, že všichni psi jsou chytří.”

Konceptuální grafy (2)

referent se skládá (viz. předchozí slide) z **kvantifikátoru** (existenciální, nebo definovaný (univerzální, kolektivní, aj.)), **designátoru** (ten označuje identifikátor instance, např. jméno) a případně tzv. **deskriptoru** (konceptuální graf popisující daný koncept).

kontext je koncept s neprázdným deskriptorem

Příklad (Kontext)



“John říká, že všichni psi jsou chytří.”

☹ není FOL

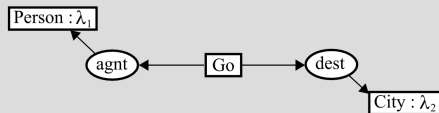
Konceptuální grafy (3)

lambda výrazy představují jakási “makra” – umožňují definovat konceptuální relace pomocí “vzoru” konceptuálního grafu. “Dosazované” proměnné se označí symboly λ_i .

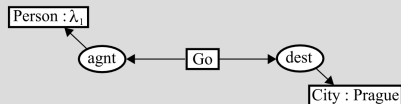
Konceptuální grafy (3)

lambda výrazy představují jakási “makra” – umožňují definovat konceptuální relace pomocí “vzoru” konceptuálního grafu. “Dosazované” proměnné se označí symboly λ_i .

Příklad (lambda výrazy)



def. binární relaci “Go”.



def. unární relaci “Go to Prague”.

Konceptuálních grafech – inference

- inference využívá několika forward chaining pravidel² (zobecnění grafu, specializace grafu, ekvivalentní úpravy).

²<http://www.jfsowa.com/cg/cgstandw.htm>

Konceptuálních grafech – inference

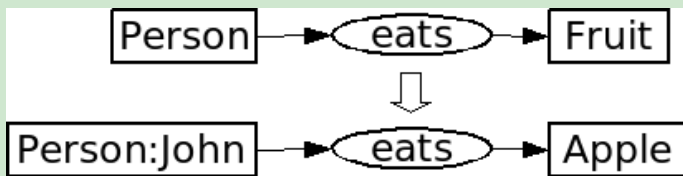
- inference využívá několika forward chaining pravidel² (zobecnění grafu, specializace grafu, ekvivalentní úpravy).
- vyhledávání (jakožto složitější inferenční procedura) se provádí pomocí tzv. **projekce**. Ta hledá výskyt vzoru konceptuálního grafu v grafu odpovídajícím dané znalostní bázi s využitím hierarchie konceptuálních typů a typů konceptuálních relací.

²<http://www.jfsowa.com/cg/cgstandw.htm>

Konceptuálních grafech – inference

- inference využívá několika forward chaining pravidel² (zobecnění grafu, specializace grafu, ekvivalentní úpravy).
- vyhledávání (jakožto složitější inferenční procedura) se provádí pomocí tzv. **projekce**. Ta hledá výskyt vzoru konceptuálního grafu v grafu odpovídajícím dané znalostní bázi s využitím hierarchie konceptuálních typů a typů konceptuálních relací.

Příklad (projekce)



²<http://www.jfsowa.com/cg/cgstandw.htm>

Konceptuální grafy – nástroje

CharGer – editor CG

(<http://sourceforge.net/projects/charger>)

Konceptuální grafy – nástroje

CharGer – editor CG

(<http://sourceforge.net/projects/charger>)

Notio – Java knihovna + API pro manipulaci s CG

(<http://backtrack.uwaterloo.ca/CG/projects/notio>)

Konceptuální grafy – nástroje

CharGer – editor CG

(<http://sourceforge.net/projects/charger>)

Notio – Java knihovna + API pro manipulaci s CG

(<http://backtrack.uwaterloo.ca/CG/projects/notio>)

Prolog+CG – inferenční stroj pro CG v Prologu

(<http://prologpluscg.sourceforge.net>)

Konceptuální grafy – nástroje

CharGer – editor CG

(<http://sourceforge.net/projects/charger>)

Notio – Java knihovna + API pro manipulaci s CG

(<http://backtrack.uwaterloo.ca/CG/projects/notio>)

Prolog+CG – inferenční stroj pro CG v Prologu

(<http://prologpluscg.sourceforge.net>)

Amine – novější verze Prolog+CG

(<http://amine-platform.sourceforge.net>)

Konceptuální grafy – nástroje

CharGer – editor CG

(<http://sourceforge.net/projects/charger>)

Notio – Java knihovna + API pro manipulaci s CG

(<http://backtrack.uwaterloo.ca/CG/projects/notio>)

Prolog+CG – inferenční stroj pro CG v Prologu

(<http://prologpluscg.sourceforge.net>)

Amine – novější verze Prolog+CG

(<http://amine-platform.sourceforge.net>)

DNA – anotační nástroj využívající pro vizualizaci CG

(<http://labe.felk.cvut.cz/uhlir/DNATWeb/DNAThome.html>)

Amine4

The screenshot shows the CG Operations Interface software. It features a menu bar (File, Edit, Font, Parameters, Ontology, CG Operations, Animation, Help) and a toolbar. The main workspace is divided into three panes:

- Input CG1:** Displays a graph with a node 'Sex = male' connected to 'Human.super' via a 'sexOf' relationship.
- Input CG2:** Displays a graph with 'Human' at the top, connected to 'Object' via 'sexOf' and 'push' relationships. 'Object' is also connected to another 'Object' via an 'on' relationship.
- Output CG:** Displays a combined graph showing the relationships from both input ontologies.

A sidebar on the left contains various operations: Equal, MaximalJoin, Generalize, Subsume, Subsume&sk, Analogy, Compare, CoveredBy, IsCanonic, Expand, and Contract. The status bar at the bottom indicates the current ontology path: `*** Current LexiconsOntology : /home/kremen/programs/Amine4/samples/ontology/ManOntologyContext.xml`

- editace/prohlížení ontologií

Amine4

The screenshot displays the CG Operations Interface, a software tool for editing and visualizing concept graphs. It features three main windows:

- Input CG1:** Shows a simple graph with a node labeled "Sex = male" connected to a node labeled "Human:super".
- Input CG2:** Shows a graph with a central node "Human" connected to "Object" via a "Push" relationship. There are also "sex" and "on" relationships between "Human" and "Object".
- Output CG:** Shows a more complex graph combining elements from the input graphs, including "Sex = male", "Human", "Object", and "Push" relationships.

The interface includes a menu bar (File, Edit, Font, Parameters, Ontology, CG Operations, Animation, Help), a toolbar with various icons, and a sidebar on the left with buttons for operations such as "Equal", "Generalize", "Subsume", "Subsume&sk", "Compare", "IsCanonic", and "Contract". The status bar at the bottom indicates the current file path: "Current LexiconsOntology : /home/kremen/programs/Amine4/samples/ontology/ManOntologyContext.xml".

- editace/prohlížení ontologií
- editace/prohlížení CG

Amine4

The screenshot displays the Amine4 software interface, titled "CG Operations Interface - Untitled". It features a menu bar (File, Edit, Font, Parameters, Ontology, CG Operations, Animation, Help) and a toolbar. The main workspace is divided into three panels:

- Input CG1:** Shows a graph with a node "Sex = male" connected to "Human.super" via a "sexOf" relationship.
- Input CG2:** Shows a graph with a central node "Human" connected to "Object" via "sexOf" and "push" relationships, and "Object" connected to "Object" via an "on" relationship.
- Output CG:** Shows a combined graph with nodes "Sex = male", "Human", and "Object", and relationships "sexOf", "push", and "on".

On the left side, there is a sidebar with various operations: Equal, Generalize, Subsume, Subsume&Skt, Compare, IsCanonic, and Contract, each with a right-pointing arrow. Below the sidebar, there are tabs for "LF Editor", "CGIF Editor", and "CG Graphic Editor". At the bottom, a status bar indicates the current file path: "Current LexiconsOntology : /home/kremen/programs/Amine4/samples/ontology/ManOntologyContext.xml".

- editace/prohlížení ontologií
- editace/prohlížení CG
- operace nad CG –
příklad: JOIN

Amine4

The screenshot displays the 'CG Operations Interface' with three main windows for editing Conceptual Graphs (CGs). The top-left window, 'Input CG1', shows a simple graph with a node 'Sex = male' connected to 'Human:super'. The top-right window, 'Input CG2', shows a graph with 'Human' connected to 'Push', which is connected to 'Object'. The bottom window, 'Output CG', shows a similar graph to 'Input CG2' but with an additional 'Sex = male' node connected to 'Human'. The interface includes a menu bar (File, Edit, Font, Parameters, Ontology, CG Operations, Animation, Help), toolbars, and a sidebar with various operations like 'Equal', 'Generalize', 'Subsume', 'SubsumeSk', 'Compare', 'IsCanonic', and 'Contract'. The status bar at the bottom indicates the current file path: 'Current LexiconsOntology : /home/kremen/programs/Amine4/samples/ontology/ManOntologyContext.xml'.

- editace/prohlížení ontologií
- editace/prohlížení CG
- operace nad CG – příklad: JOIN
- inference CG+Prolog

Amine4

The screenshot displays the 'CG Operations Interface' software. It features a menu bar (File, Edit, Font, Parameters, Ontology, CG Operations, Animation, Help) and a toolbar. The interface is divided into several panes:

- Input CG1:** Shows a graph with a node 'Sex = male' connected to 'Human:super' via a 'subOf' relationship.
- Input CG2:** Shows a graph with 'Human' connected to 'Push' via 'subOf', and 'Push' connected to 'Object' via 'on'.
- Output CG:** Shows a combined graph with 'Sex = male' connected to 'Human' via 'subOf', and 'Human' connected to 'Push' via 'subOf'. 'Push' is also connected to 'Object' via 'on'.

On the left side, there is a vertical menu with various operations: Equal, Generalize, Subsume, SubsumeSk, Compare, IsCanonic, Contract, MaximalJoin, and Analogy. The bottom status bar indicates the current ontology path: 'Current LexiconsOntology : /home/kremen/programs/Amine4/samples/ontology/ManOntologyContext.xml'.

- editace/prohlížení ontologií
- editace/prohlížení CG
- operace nad CG – příklad: JOIN
- inference CG+Prolog
- multiagentní systémy

Konceptuální grafy – shrnutí

- (J.F. Sowa 80's) jsou představitelem formálních (strojově zpracovatelných) a přitom dobře čitelných, intuitivních jazyků

Konceptuální grafy – shrnutí

- (J.F. Sowa 80's) jsou představitelem formálních (strojově zpracovatelných) a přitom dobře čitelných, intuitivních jazyků
- vycházejí myšlenkově z Pierceho existenciálních grafů [Sow00], [Dau01].

Konceptuální grafy – shrnutí

- (J.F. Sowa 80's) jsou představitelem formálních (strojově zpracovatelných) a přitom dobře čitelných, intuitivních jazyků
- vycházejí myšlenkově z Pierceho existenciálních grafů [Sow00], [Dau01].
- jsou expresivnější než logika prvního řádu – nerozhodnutelnost !!!

Konceptuální grafy – shrnutí

- (J.F. Sowa 80's) jsou představitelem formálních (strojově zpracovatelných) a přitom dobře čitelných, intuitivních jazyků
- vycházejí myšlenkově z Pierceho existenciálních grafů [Sow00], [Dau01].
- jsou expresivnější než logika prvního řádu – nerozhodnutelnost !!!
- předchozí problém řeší tzv. *simple graphs* (J.F. Sowa 80's), které omezují tvar referentu a neumožňují tvořit kontexty.