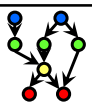


Lecture 18 of 42

Knowledge Representation Concluded: KE, CIKM, & Representing Events over Time Discussion: Structure Elicitation, Event Calculus

William H. Hsu
Department of Computing and Information Sciences, KSU

KSOL course page: http://snipurl.com/v9v3
Course web site: http://www.kddresearch.org/Courses/CIS730
Instructor home page: http://www.cis.ksu.edu/~bhsu

Reading for Next Class:
Section 10.4 – 10.6, p. 341 – 353, Russell & Norvig 2nd edition
IM: http://en.wikipedia.org/wiki/Information_management
Event calculus: http://en.wikipedia.org/wiki/Event_calculus
Protégé-OWL tutorial: http://bit.ly/3rM1pB

Lecture Outline

- Reading for Next Class: Sections 10.4 – 10.6 (p. 341 – 353), R&N 2e
- Last Class: Knowledge Engineering (KE), Protocol Analysis, Fluents
  - Ontology engineering: defining classes/concepts, slots
  - Concept elicitation techniques
    - Unstructured
    - Structured
    - Protocol analysis
- Today: Event and Fluent Calculi, CIKM
  - Representing time, events: from situation calculus to event, fluent calculi
  - Knowledge acquisition (KA) and capture
  - Computational information and knowledge management (CIKM)
- Coming Week: CIKM, Logical KR Concluded; Classical Planning

Acknowledgements

Natasha Noy
Senior Research Scientist
BMIR

Samson Tu
Senior Research Scientist
BMIR

Holger Knublauch
Vice President, TopQuadrant
previously Research Fellow,
Stanford Medical Informatics & Univ. of Manchester

Georghe Tecuci
Professor of Computer Science
Director, Learning Agents Center
George Mason University

Milos Hauskrecht
Associate Professor of Computer Science
University of Pittsburgh

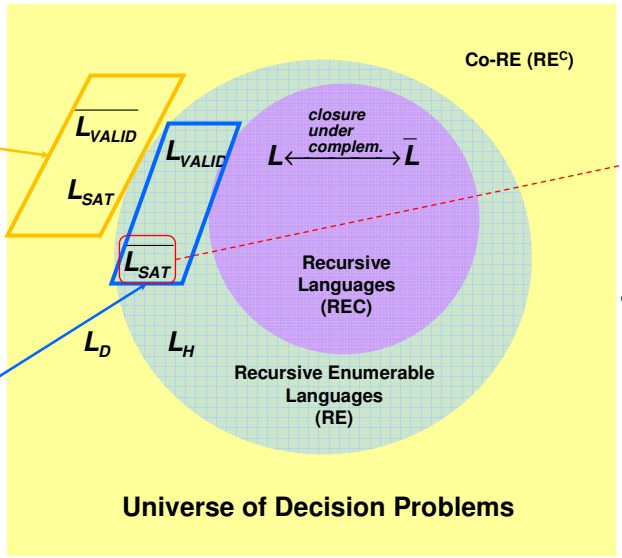
Decision Problems: Review

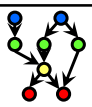
Undecidable duals
$\alpha \in L_{VALID}{}^{C}$ iff
$\neg\alpha \in L_{SAT}$

L_H: Halting problem

L_D: Diagonal problem

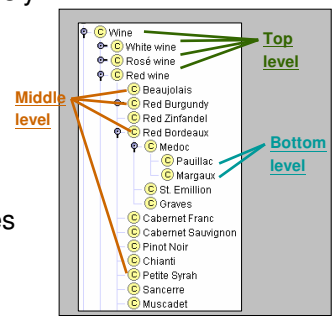
Semi-decidable duals:
$\alpha \in L_{VALID}$ iff
$\neg\alpha \in L_{SAT}{}^{C}$





Concepts/Classes: Review

- "Concept" and "Class" are used synonymously
- Class: concept in the domain
  - wines
  - wineries
  - red wines
- Collection of elements with similar properties
- Instances of classes
  - Particular glass of California wine



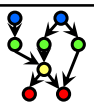
Slots/Attributes/Relations: Review

- Slots in class definition *C*
  - *Describe attributes of instances of C*
  - *Describe relationships to other instances*
  - *e.g., each wine will have color, sugar content, producer, etc.*
- Property constraints (facets): describe/limit possible values for slot

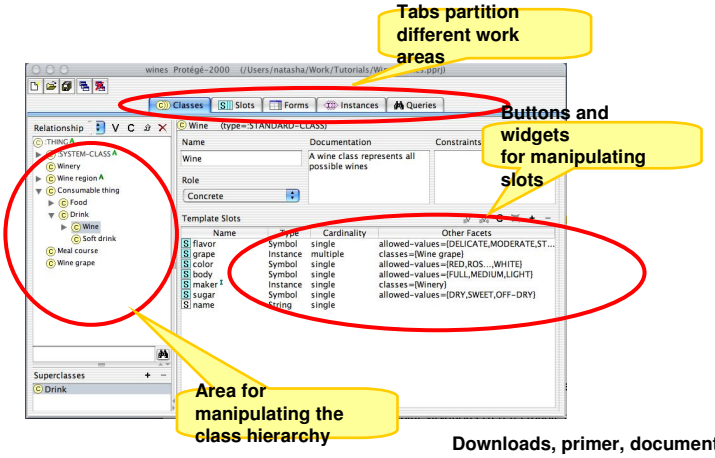
Template Slots

Name	Type	Cardinality	Other Facets
body	Symbol	single	allowed-values={FULL,MEDIUM,LIGHT}
color	Symbol	single	allowed-values={RED,ROSÉ,WHITE}
flavor	Symbol	single	allowed-values={DELICATE,MODERATE,STRONG}
grape	Instance	multiple	classes={Wine grape}
maker	Instance	single	classes={Winery}
name	String	single	
sugar	Symbol	single	allowed-values={DRY,SWEET,OFF-DRY}

Slots & facets for Concept/Class *Wine*



Protégé – Default Interface: Review



Downloads, primer, documentation:
http://protege.stanford.edu

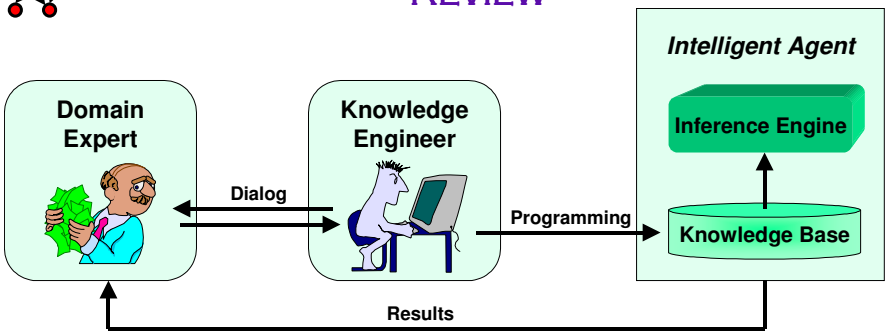
Adapted from slides © 2005 N. Noy & S. Tu
Stanford Center for Biomedical Informatics Research
http://bmir.stanford.edu

Knowledge Engineering: Review

- A scenario for manual knowledge acquisition
- Elicitation of expert's conception of a domain
- Elicitation based on the personal construct theory
- Knowledge acquisition for role-limiting methods
- Advanced approaches to KB and agent development

© 2001 G. Tecuci, George Mason University
CS 785 Knowledge Acquisition and Problem-Solving http://lalab.gmu.edu/cs785/

How Agents Are Built: Review



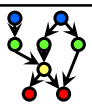
A knowledge engineer attempts to understand how a subject matter expert reasons and solves problems and then encodes the acquired expertise into the agent's knowledge base.

The expert analyzes the solutions generated by the agent (and often the knowledge base itself) to identify errors, and the knowledge engineer corrects the knowledge base.

Agent Development Process: Review

- Defining problem to solve and system to be built: requirements specification
- Understanding the expertise domain
- Choosing or building an agent building tool: Inference engine and representation formalism
- Development of the object ontology
- Development of problem solving rules or methods
- Refinement of the knowledge base

Feedback loops among all phases



Elicitation Methodology: Review

(based primarily on Gammack, 1987)

1. **Concept elicitation: methods**
   (elicit concepts of domain, *i.e.* agreed-upon vocabulary)
2. **Structure elicitation: card-sort method**
   (elicit some structure for concepts)
3. **Structure representation**
   (formally represent structure in semantic network)
4. **Transformation of representation**
   (transform representation to be used for some desired purpose)

CS 785 Knowledge Acquisition and Problem-Solving http://lalab.gmu.edu/cs785/

Actions, Situations, Time & Events [1]: Situation Calculus Revisited

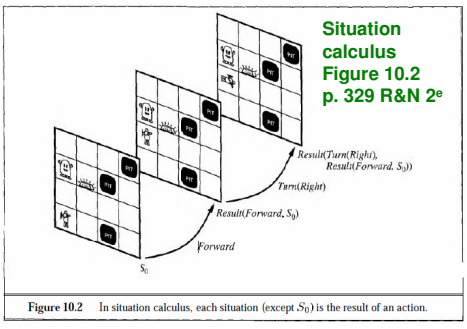
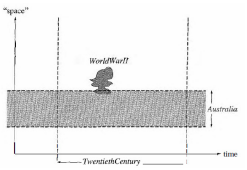


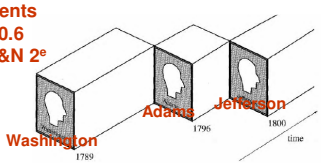
Figure 10.2 In situation calculus, each situation (except S_0) is the result of an action.

Situation calculus
Figure 10.2
p. 329 R&N 2e



Event calculus
Figure 10.3
p. 336 R&N 2e

State fluents
Figure 10.6
p. 340 R&N 2e



- Event Calculus: Reasoning about Entities over Time, Space
- Fluent Calculus: Variant of Situation Calculus
  - Conditions (predicates) that can change over time
  - Defaults
  - ∘ (concatenation)

Actions, Situations, Time & Events [2]: Event Calculus

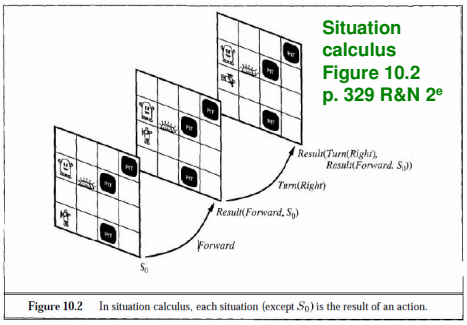
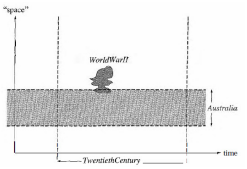


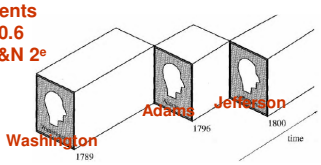
Figure 10.2 In situation calculus, each situation (except S_0) is the result of an action.

Situation calculus
Figure 10.2
p. 329 R&N 2e



Event calculus
Figure 10.3
p. 336 R&N 2e

State fluents
Figure 10.6
p. 340 R&N 2e



Figures © 2003 S. Russell & P. Norvig. Reused with permission.

- Event Calculus: Reasoning about Entities over Time, Space
- Fluent Calculus: Variant of Situation Calculus
  - Conditions (predicates) that can change over time
  - Defaults
  - ∘ (concatenation)

Actions, Situations, Time & Events [3]: Fluent Calculus

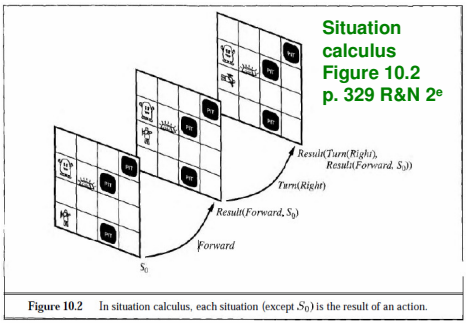
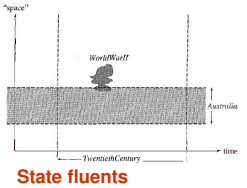


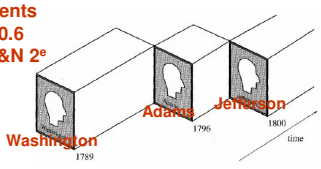
Figure 10.2 In situation calculus, each situation (except S_0) is the result of an action.

Situation calculus
Figure 10.2
p. 329 R&N 2e



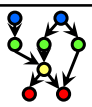
Event calculus
Figure 10.3
p. 336 R&N 2e

State fluents
Figure 10.6
p. 340 R&N 2e



Figures © 2003 S. Russell & P. Norvig. Reused with permission.

- Event Calculus: Reasoning about Entities over Time, Space
- Fluent Calculus: Variant of Situation Calculus
  - Conditions (predicates) that can change over time
  - Defaults
  - ∘ (concatenation)

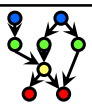


Preview: Computational Information & Knowledge Management

- Information Management
  - Data acquisition: instrumentation, collection, polling, elicitation
  - Data and information integration: combining multiple sources
    - ⇨ May be heterogeneous (different in quality, format, rate, *etc.*)
    - ⇨ Underlying formats, properties may correspond to different ontologies
    - ⇨ Ontology mappings (functions to convert between ontologies) needed
  - Data transformation: preparation for reasoning, learning
    - ⇨ Preprocessing
    - ⇨ Cleaning
  - Includes knowledge capture: assimilation from various sources
- Knowledge Management
  - Term used most often in business administration, management science
  - Related to IM, but capability and process-centered
  - Focus on learning and KA, organization theory, decision theory
    - ⇨ Discussion, apprenticeship, forums, libraries, training/mentoring
    - ⇨ Modern theory: KBs, Expert Systems, Decision Support Systems

Terminology

- Knowledge Engineering (KE): Process of KR Design, Acquisition
  - Knowledge
    - ⇨ What agents possess (epistemology) that lets them reason
    - ⇨ Basis for rational cognition, action
    - ⇨ Knowledge gain (acquisition, learning): improvement in problem solving
  - Knowledge level (*vs.* symbol level): level at which agents reason
  - Semantic network: inheritance and membership/containment relationships
  - Knowledge elicitation: KA/KE process from human domain experts
    - ⇨ Protocol analysis: preparing, conducting, interpreting interview
    - ⇨ Less formal methods: subjective estimation & probabilities
- Fluents: Conditions (Predicates) That Can Change over Time
  - Classes, nominals (objects / class instances): spatial, temporal extent
  - Fluent calculus: situation calculus with defaults, ∘ (concatenation)
- Computational Information and Knowledge Management (CIKM)
  - Data/info integration & transformation: collecting, preparing data
  - Includes knowledge capture: assimilation from various sources



Summary Points

- Last Class: Prolog in Brief, Description Logics, Ontologies
  - Prolog examples
  - Ontologies: formal languages for describing domains for KR
  - KR as basis of learning and reasoning
  - $\mathcal{ALC}$, $\mathcal{SHOIN}$, and Web Ontology Language (OWL)
- Today: More Ontology Design; Knowledge Engineering, Elicitation
  - Concept elicitation techniques
    - Unstructured
    - Structured
    - Protocol analysis
  - Knowledge acquisition (KA); info and knowledge management defined
  - Situation calculus revisited; time and event calculus, fluent calculus
- Next Class: More KE, Semantic Nets
  - KA and knowledge capture: elicitation concluded (structure elicitation)
  - Computational information and knowledge management (CIKM)
- Coming Week: Logical KR Concluded; Planning