



Computational Logic: Mix

Maria Spichkova

spichkov@in.tum.de

09.07.2008



Agenda

- „Computational Logic“ Program : @ TU Dresden
- Cognitive Robotics
 - Situation Calculus
 - Fluent Calculus
 - Event Calculus
 - FLUX
- ACE



„Computational Logic“ Program

- Propositional logic
- FOL
- Advanced Logic
 - higher-order logics
 - temporal logic
 - mu-calculus
 - schematic tableaux
 - ...
- Deduction, induction
- Knowledge representation and reasoning
- ProLog, ProLog, ProLog...



- High-level robotics/control:
 - Planning
 - Interface
 - Execution monitoring
- Out of scope (low level):
 - Localisation
 - Vision System
 - Sensors
 - Gripper, motor...
 - ...



Cognitive Robotics: AI

- Localisation
- Navigation
- Map learning
- Planning with incomplete information
- Problem categories:
 - Prediction
 - Postdiction (explanation)
 - Planning
 - Verification



Situation Calculus

- Representing and reasoning about dynamical domains
- Introduced by John McCarthy in 1963
- Represents changing scenarios as a set of FOL formulae

- Fluent – atomic state property
- State – defined with the fluents that holds
- Situation – sequence of actions performed so far



Situation Calculus: GOLOG (aLGOL in LOGic)

- Agent programming language GOLOG
- User provides axioms about:
 - the initial situation
 - the preconditions and effects of actions
 - general properties
- User controls the search effort
- Needed: Complete information about the initial state



Situation Calculus: Basic Elements

- Sorts
 - Predef.: FLUENT, ACTION, SIT, BOOL
 - Userdef.: S
- Function and Predicate Symbols
 - $S_0 : SIT$
 - $Do : ACTION \times SIT \rightarrow SIT$
 - $Holds : FLUENT \times SIT \rightarrow BOOL$
 - $Poss : ACTION \times SIT \rightarrow BOOL$
 - $< : SIT \times SIT \rightarrow BOOL$
 - User defined: over S, $S \rightarrow FLUENT$, $S \rightarrow ACTION$, ...



Situation Calculus: Axioms, Rules

- Action precondition axioms
- Successor state axioms
 - Positive effect axioms
 - Negative effect axioms
- Axioms describing the world in various situations
- Foundational axioms



Situation Calculus: Example

- Action precondition axiom:

$$\text{Holds}(\text{Open}(r), s) \wedge \text{Holds}(\text{InFrontOf}(r), s) \rightarrow \\ \text{Poss}(\text{Enter}(r), s)$$

- Positive effect axiom:

$$\text{Poss}(a, s) \wedge a = \text{Enter}(r) \rightarrow \\ \text{Holds}(\text{InRoom}(r), \text{Do}(a,s))$$

- Negative effect axiom:

$$\text{Poss}(a, s) \wedge \exists r'. a = \text{Enter}(r') \wedge r \neq r' \rightarrow \\ \neg \text{Holds}(\text{InRoom}(r), \text{Do}(a,s))$$



Fluent Calculus: Basic Elements

- TU Dresden
- Sorts
 - Predef.: FLUENT, ACTION, STATE, SIT, BOOL
 - Userdef.: S
- Function and Predicate Symbols
 - $S_0 : \text{SIT}$, $\emptyset : \text{STATE}$, $\text{State} : \text{SIT} \rightarrow \text{STATE}$
 - $\text{Do} : \text{ACTION} \times \text{SIT} \rightarrow \text{SIT}$, $\circ : \text{STATE} \times \text{STATE} \rightarrow \text{STATE}$
 - $\text{Holds} : \text{FLUENT} \times \text{STATE} \rightarrow \text{BOOL}$
 - $\text{Poss} : \text{ACTION} \times \text{STATE} \rightarrow \text{BOOL}$
 - $< : \text{SIT} \times \text{SIT} \rightarrow \text{BOOL}$
 - User defined ...



Fluent calculus: What is different?

- Argumentation about states
 - FLUENT is a subsort of STATE
- Axioms:
 - Precondition axiom (for every action)
 - State update axiom (for every action)
- Successor state axioms:
 - Situation calculus: goes through each fluent
 - Fluent calculus: goes through each action

Also: in FLUX



Event Calculus

- Global time line
- Event ~ action of a robot
- Happens(e, t)

Happens(Enter(Room1, t) \wedge t > 16 \wedge t < 20

- HoldsAt(f, t)
- Initiates(e, f, t) and Terminates(e, f, t)
- Initially(f)
- Clipped(t₁, f, t₂) and Declipped(t₁, f, t₂)

Also: in FLUX



- An Action Programming Language
- ProLog based

```
is_room(R) :- R=r1; R=r2; R=r3.
```

...

```
is_fluent(in_room(R)) :- is_room(R).
```

...

```
is_action(enter(R)) :- is_room(R).
```

...

```
poss(enter(R), Z) :- holds(at_door(D), Z),  
    holds(in_room(R1), Z), connected(D, R1, R),  
    not_holds(closed(D), Z).
```



FLUX: Planning

init(Z0), exec(Z0, s0, Z, S, 10),
holds(in_room(r3), Z)

Z0 = ...

S = do(..., do(..., s0)...)

Z = ...



FLUX: More interesting stuff

- Causality: macro Causes
- Concurrent Actions
- Sensing and Knowledge
 - robot knows that f holds in situation s
if all possible states in s satisfy f
- Different kinds of planning
- ...



Attempto Controlled English

- University of Zurich
- Controlled natural language (controlled subset)
- Tools
 - Attempto Parsing Engine APE
 - Attempto Reasoner RACE
 - Attempto Verbaliser DRACE
 - ...



- ACE texts
 - computer-processable
 - can be translated into [FOL](#), [FLUX](#), [RuleML](#), ...
- The vocabulary of ACE
 - predefined function words (e.g. determiners, conjunctions, prepositions), and some predefined fixed phrases
 - content words (nouns, proper names, verbs, adjectives, adverbs) [basic lexicon of content words]
- Structure of a simple ACE sentence:
subject + verb + complements + adjuncts



ACE: Tiny examples

- A simple sentence describes that something is the case — a fact, an event, a state.
 - The temperature is -2°C .
 - A customer inserts 2 cards.
 - At least 3 cards and exactly 2 codes are valid.
- A trusted customer inserts a valid card manually.
 - Who inserts a card?
 - Which customer inserts a card?
 - What does a customer insert?
 - How does a customer insert a card?



Questions?