The Bari Lectures

Tuesday:
Computational Semantics in Spoken Dialogue Systems

Wednesday:
Generating Speech Recognition Packages

Thursday:
Implementing Spoken Dialogue Systems

Johan Bos

Language Technology Group

The University of Edinburgh

Computational Semantics in Spoken Dialogue Systems

Johan Bos

Language Technology Group

The University of Edinburgh

Aims of this Talk

- Next generation of spoken dialogue system will go beyond simple slot-filling
- Novel techniques from computational semantics can play a crucial role here
- First-order model building lends itself naturally for many interpretation tasks
- Illustrate the idea with an architecture used in two prototype implementations

What we will learn today

- What kind of inference techniques are out there and how do we use them
- What are current methods of semantic interpretation in dialogue systems, and what are their shortcomings
- How can we improve this, using formal methods in practical systems

Outline of this Talk

- Part I: Foundations
  - Tools for Semantic Interpretation
  - Choice of logic, model building
- Part II: From Natural Language to Logic
  - Ambiguities
  - Discourse Representation Structures
- Part III: Applications:
  - Controlling devices in Home Automation
  - Instructing Mobile Robots

Part I: Foundations

- Let’s first consider current technology for semantic interpretation (slot filling)
- Then let’s be serious about semantic interpretation and answer the following questions:
  - Which **semantic formalism** are you going to use?
  - Which **tools for interpretation** are out there?
  - How are you going to **construct representations** for expressions of natural language and deal with ambiguities?

Slot-filling (frames) for semantic interpretation

- Domain specific method, trying to find values for previously established slots
- Example: flight scheduling
  - Destination
  - Starting location
  - Time of travelling
  - Single or return flight
  - …

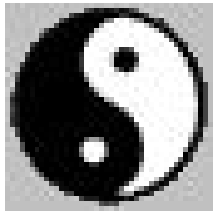
Slot-Filling (2)

- Example utterance: “I would like to fly from Rome to Pisa next Monday”
  - Destination: rome
  - Origin: pisa
  - Day: next(Monday)
- Good: missing values drive dialogue
- Bad: no natural way of dealing with more complex cases (negation, conditionals, disjunction)

Tools for Semantic Interpretation

- **Theorem Proving**
  - Useful for drawing inferences, such as checking for inconsistencies or informativeness
- **Model Building** (Model Generation)
  - Useful for checking consistency and building a discourse model
- **Model Checking**
  - Useful for querying properties of the constructed discourse model

The Yin and Yang of Inference



- **Theorem Proving** and **Model Building** function as *opposite forces*
- Suppose φ, a logical formula, representing a certain discourse δ
  - If a theorem prover succeeds in finding a **proof** for ¬φ, then δ is **inconsistent**
  - If a model builder succeeds to construct a **model** for φ, then δ is **consistent**

Using Model Building

- Example: “I want to fly from Stansted to Paris”
- Formula (First-order logic):
  - $\exists e(fly(e) \& agent(e,i) \& from(e,stansted) \& to(e,paris))$
- Axioms (travel domain)
  - $\forall x \forall e \forall z(fly(e) \& agent(e,x) \& to(e,z) \rightarrow destination(x,z))$
  - $\forall x \forall e \forall z(fly(e) \& agent(e,x) \& from(e,z) \rightarrow origin(x,z))$
  - And so on…
- Model (D the domain, F the interpretation function):
  - D={d1,d2,d3}
  - F(i)=d1, F(stansted)=d2, F(paris)=d3, F(destination)={d1,d2)}, F(origin)={d1,d3},….

Model Checking

- A Model Checker (for FOL) is a tool that checks whether a certain model satisfies certain propositions
- Almost like asking a yes-no question
  - Example: are ‘walkers’ the same as persons?
  - Query: satisfy(∀x(walk(x)↔person(x)),M,[]).
- Can also be used to extract information
- Similar to asking a wh-question
  - Example: who is a ‘walker’?
  - Query: satisfy(walk(x),M,[g(x,Answer)]).

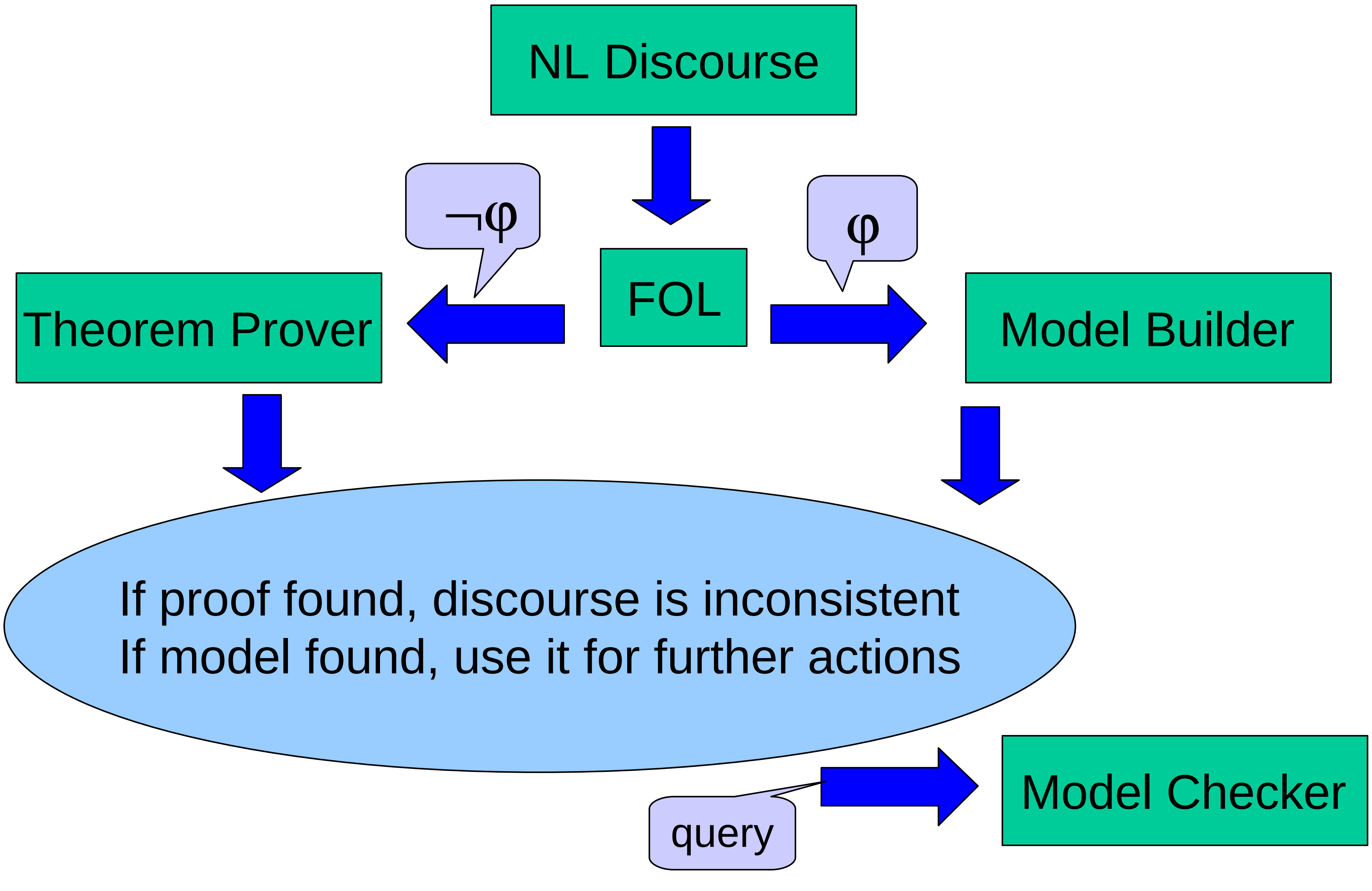
The Beauty of Finite Models

- Minimal (no redundant information)
- Flat (no recursion)
- Deals naturally with quantification, disjunction, conditionals, negation
  - "I want to fly from either Stansted or Luton, but not from Stansted on Fridays"
- Model Checking tools available
- Useful for many NLP tasks:
  - Question answering
  - Disambiguation
  - Interpretation of Instructions

Now that we know what tools are available, what is a sensible choice for semantic formalism?

- First-order logic
  - A lot of tools out there, but relatively bad computational properties
- Higher-order logic
  - Very expressive, but currently no useful inference tools
- Description logics
  - Relatively good computational properties, but limited expressive power

The picture so far…

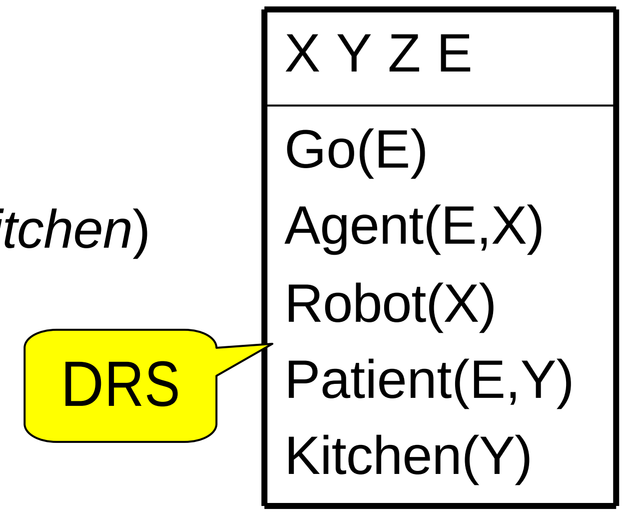


Part II: From Natural Language to Logic

- Ambiguities in Natural Language
- Discourse Representation Theory (DRT)
- Building Underspecified Discourse Representation Structures
  - Hole Semantics (Scope ambiguities)
  - Van der Sandt's “Binding and Accommodation Theory” (Anaphora, Presupposition)
- From DRT to FOL

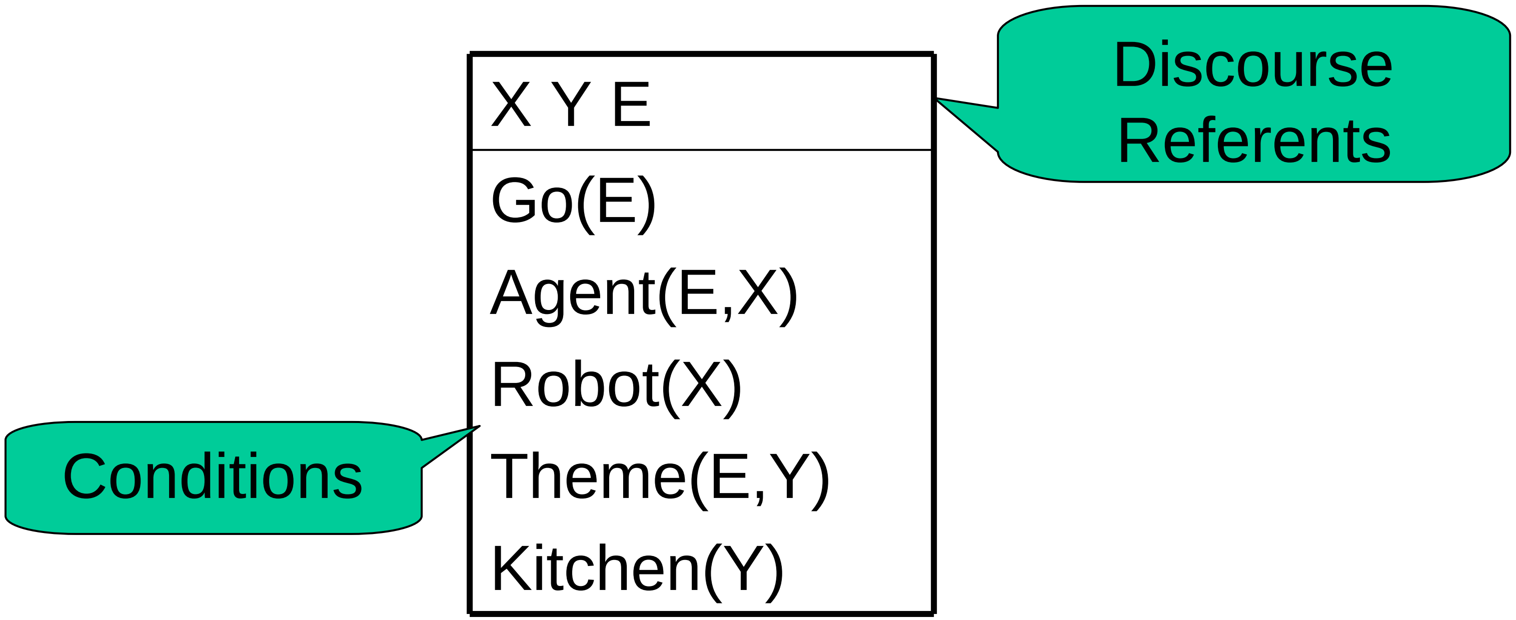
Discourse Representation Theory

- Ongoing Text (or indeed Dialogue) is represented as a Discourse Representation Structure (DRS)
  - Based on Discourse Representation Theory
  - Kamp & Reyle 1993 (big blue book)
- A DRS represents discourse entities by discourse referents and their properties
- Context Resolution
  - Anaphora (*it*, *there*, *that*)
  - Definite descriptions (*Ewan's office, the kitchen*)
  - Presuppositions (*another room*)
- First-Order Inference
  - Translate DRS to First-Order Logic
  - Model Building vs. Theorem Proving



A simple example (1/2)

- DRS for “Go to the kitchen”



A simple example (2/2)

- DRS for:

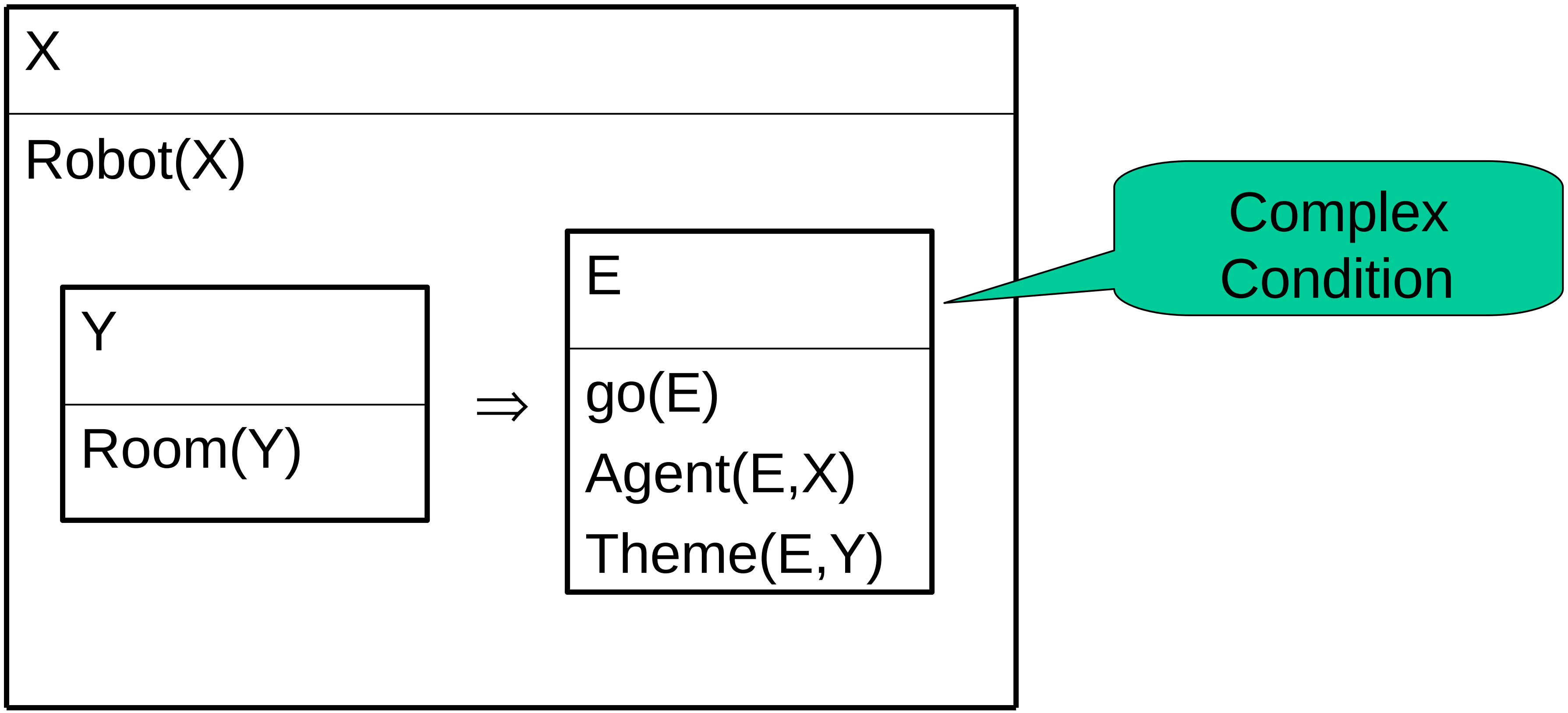
  "Go to the kitchen.
  Clean it. "

X Y E E' Z
Go(E)
Agent(E,X)
Robot(X)
Theme(E,Y)
Kitchen(Y)
Clean(E')
Agent(E',X)
Theme(E',Z)
Z=Y

Anaphoric link

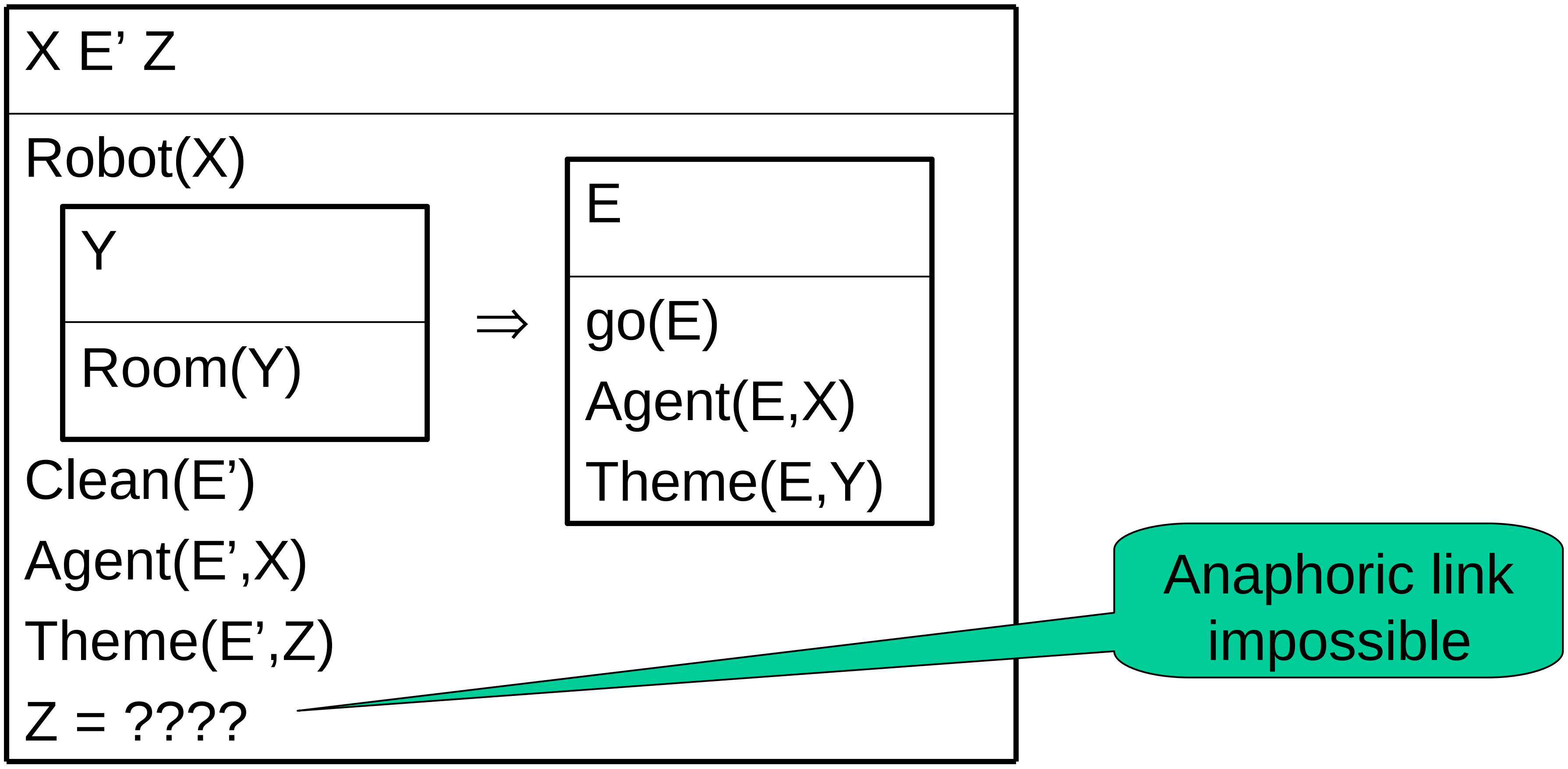
An example with conditionals (1/2)

- DRS for “Go to every room”



An example with conditionals (2/2)

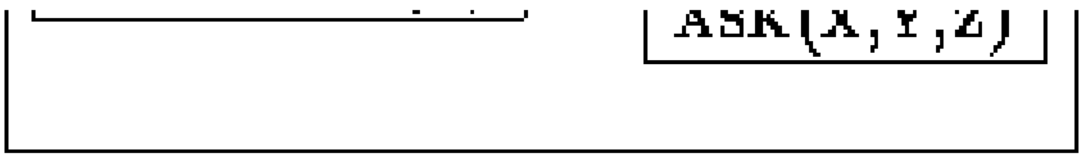
- DRS for “Go to every room. Clean it.”



From DRSs to First-Order

Example: ***A robber asked every customer some money***

$$\exists x(\text{ROBBER}(x) \wedge \forall y(\text{CUSTOMER}(y) \rightarrow \exists z(\text{MONEY}(z) \wedge \text{ASK}(x,y,z))))$$



Ambiguities in Language

- Ambiguities in expressions allow different interpretations or meanings
- Various knowledge sources help to disambiguate phrases (context, grammar, intonation, common-sense knowledge)
- Phenomena that give rise to ambiguities
  - scope, anaphora, presupposition

Scope Ambiguities

- Relative scope assignments of "*every week*" and "*a cyclist*":
  - *Every week a cyclist is hit by a bus in Bologna*.

Scope Ambiguities

- Relative scope assignments of "*every week*" and "*a cyclist*":
  - *Every week a cyclist is hit by a bus in Bologna*.
  - *He doesn't appreciate it very much*.

Scope Ambiguities

- Relative scope assignments of “*every week*” and “*a cyclist*”:
  - *Every week a cyclist is hit by a bus in Bologna*.
  - *He doesn’t appreciate it very much*.

- Structurally different semantic representations:
  - " **x(week(x)** ® $ **y(cyclist(y)&…..))**
  - $ **y(cyclist(y)&** " **x(week(x)** ® **…..))**

Anaphoric Ambiguities

- Relational noun "*part*" (implicitly anaphoric)
  - Tim: *Where were you born?*
  - Kim: *America.*
  - Tim: *Which part?*
  - Kim: *All of me, of course.*

- Different Semantic Representations:
  - **…(part(x,y)&y=america)…**
  - **…(part(x,y)&y=kim)...**

Presuppositional Ambiguities

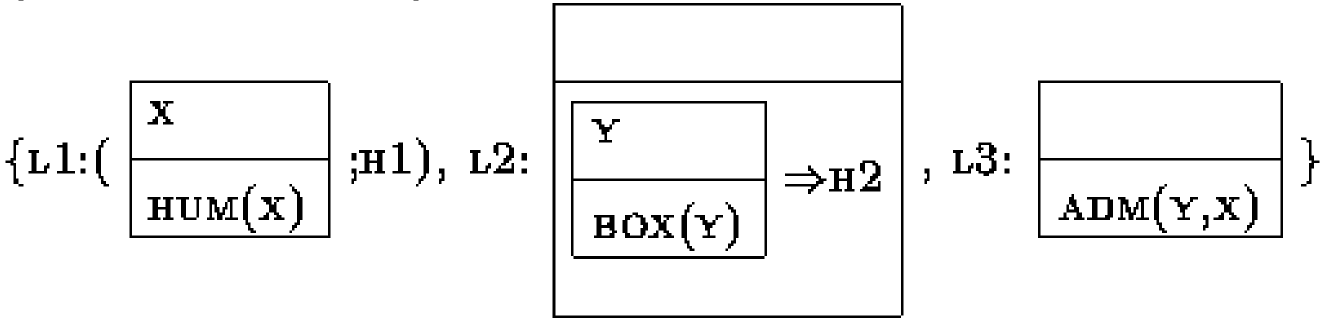
- The expression “John’s car” presupposes that <u>John has a car</u>
- The sentence “Every man likes his car” can mean:
  - Every man(1) has a car(2) and he(1) likes it(2)
  - Every man that has a car(1) likes it(1)
  - Every man(1) likes the car of him(2)

Hole Semantics

Example: ***Someone is admired by every boxer***

Dom: {L1,L2,L3,H0,H1,H2}

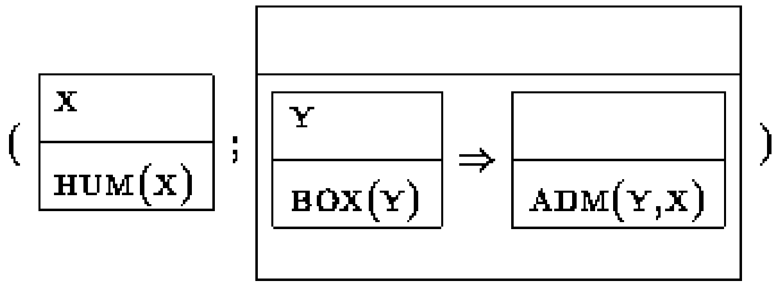
Lab:



Con: {L1 $\leq$ H0, L2 $\leq$ H0, L3 $\leq$ H1, L3 $\leq$ H2}

Plugging: P_1(H0)=L1, P_1(H1)=L2, P_1(H2)=L3

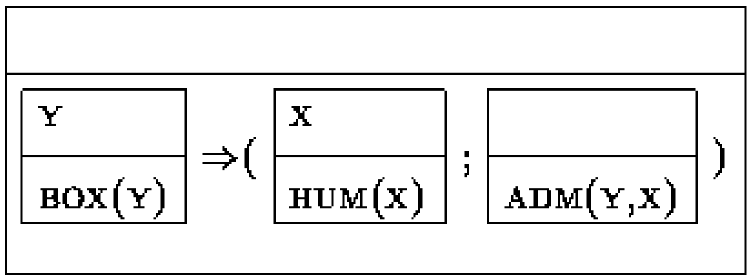
DRS:



Someone receives wide scope

Plugging: P_2(H0)=L2, P_2(H1)=L3, P_2(H2)=L1

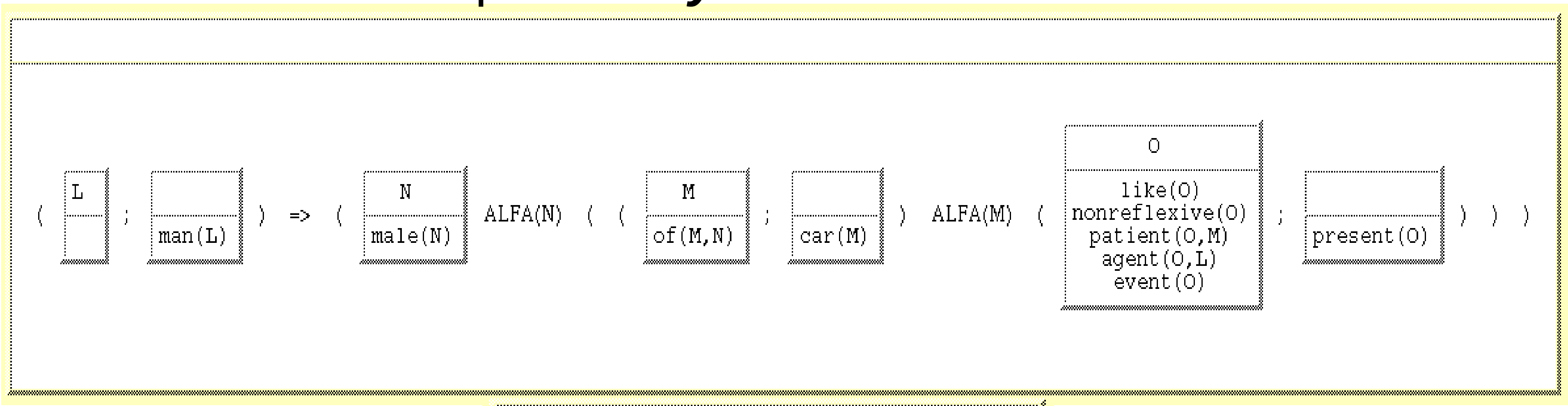
DRS:

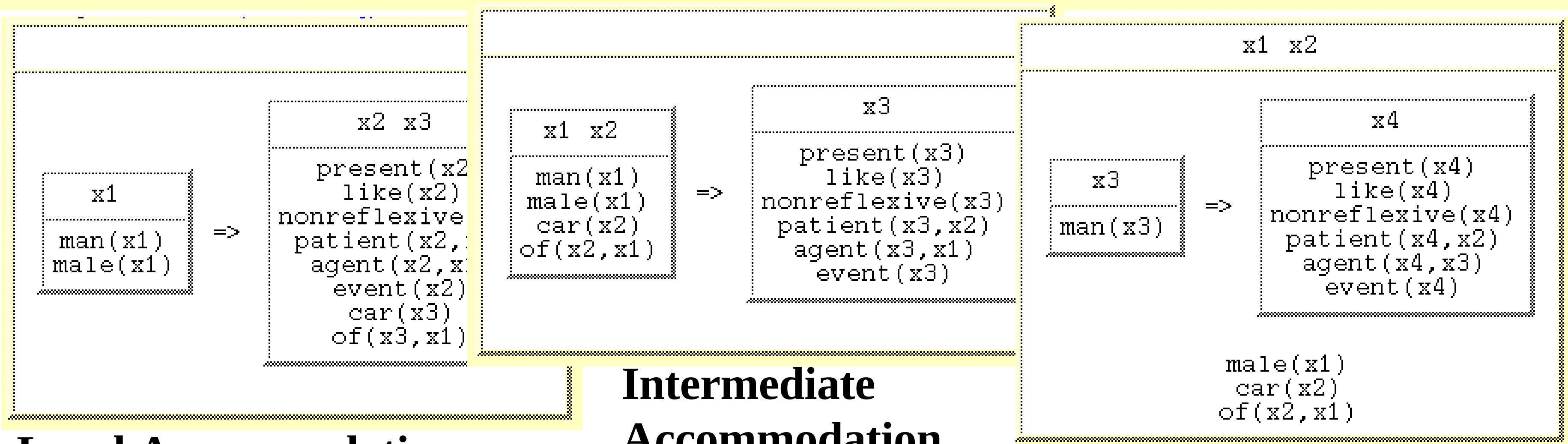


Every boxer receives wide scope

Presupposition Projection

Example: ***Every man likes his car***





Local Accommodation

Intermediate Accommodation

Global Accommodation

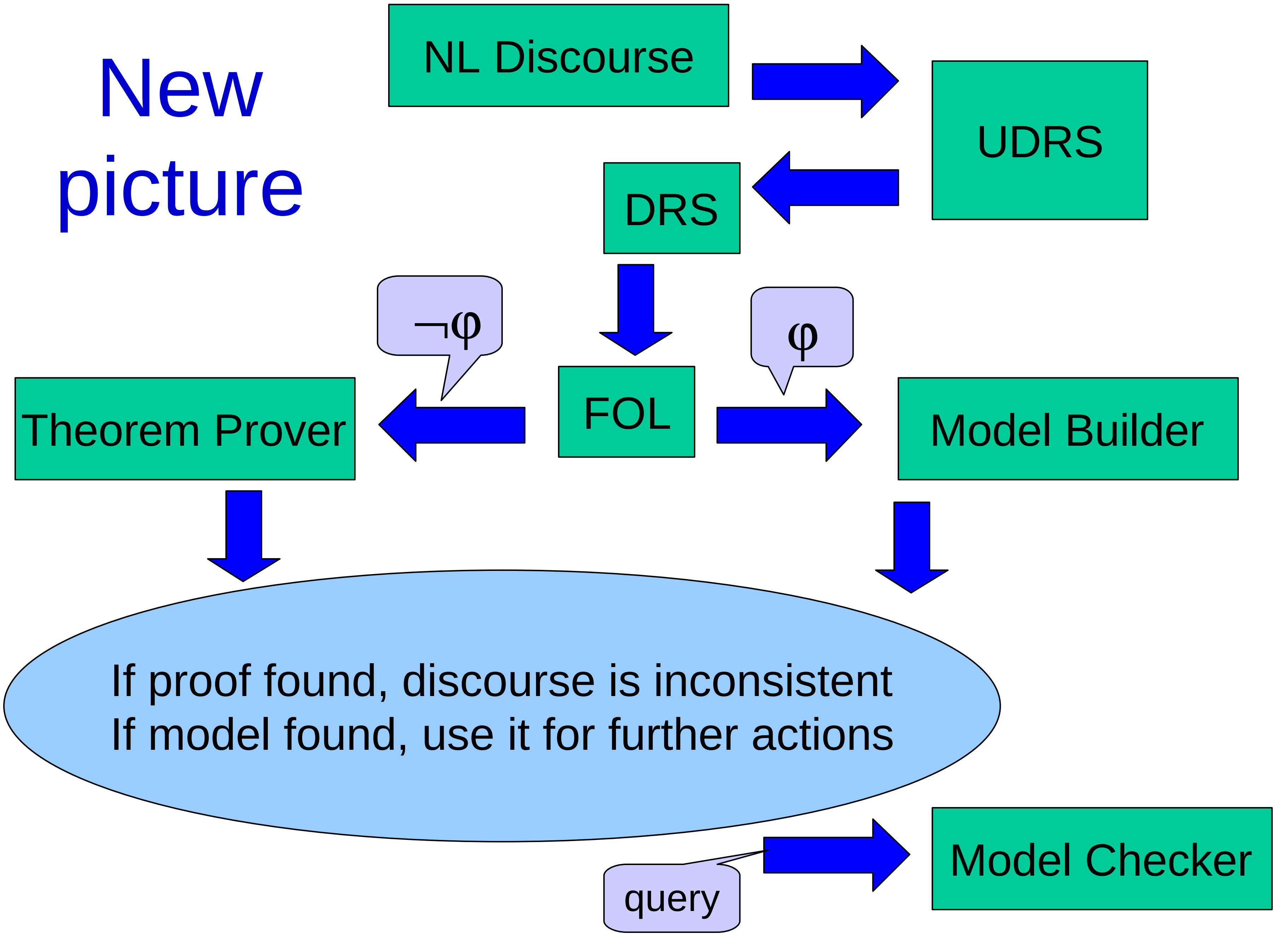
Anaphora and Presupposition (Why DRT makes a difference)

- Example 1 (Home Automation)
  - Turn on a light. Turn it off.
    Turn on another light.
- Example 2 (Mobile Robot)
  - Go to the kitchen. Stop going to the hallway.
- Both examples can be dealt with by lexical specification of presuppositions

Presuppositional Accommodation Revisited

- Example:
  - Situation: 4 lights on, 1 light off
  - Instruction: “Switch off every light!”
- AI-type of reasoning:
  - action switch-off(x) has as pre-condition on(x)
  - Problem: this will lead to an inconsistent state
- Presupposition approach:
  - Treat switch-off as an anaphoric presupposition
  - Solves the problem by virtue of intermediate (‘restrictor’) accommodation
  - Paraphrased DRS: “Switch off every light that is on”

New picture



NL Discourse
UDRS
DRS
¬φ
φ
Theorem Prover
FOL
Model Builder
If proof found, discourse is inconsistent
If model found, use it for further actions
query
Model Checker

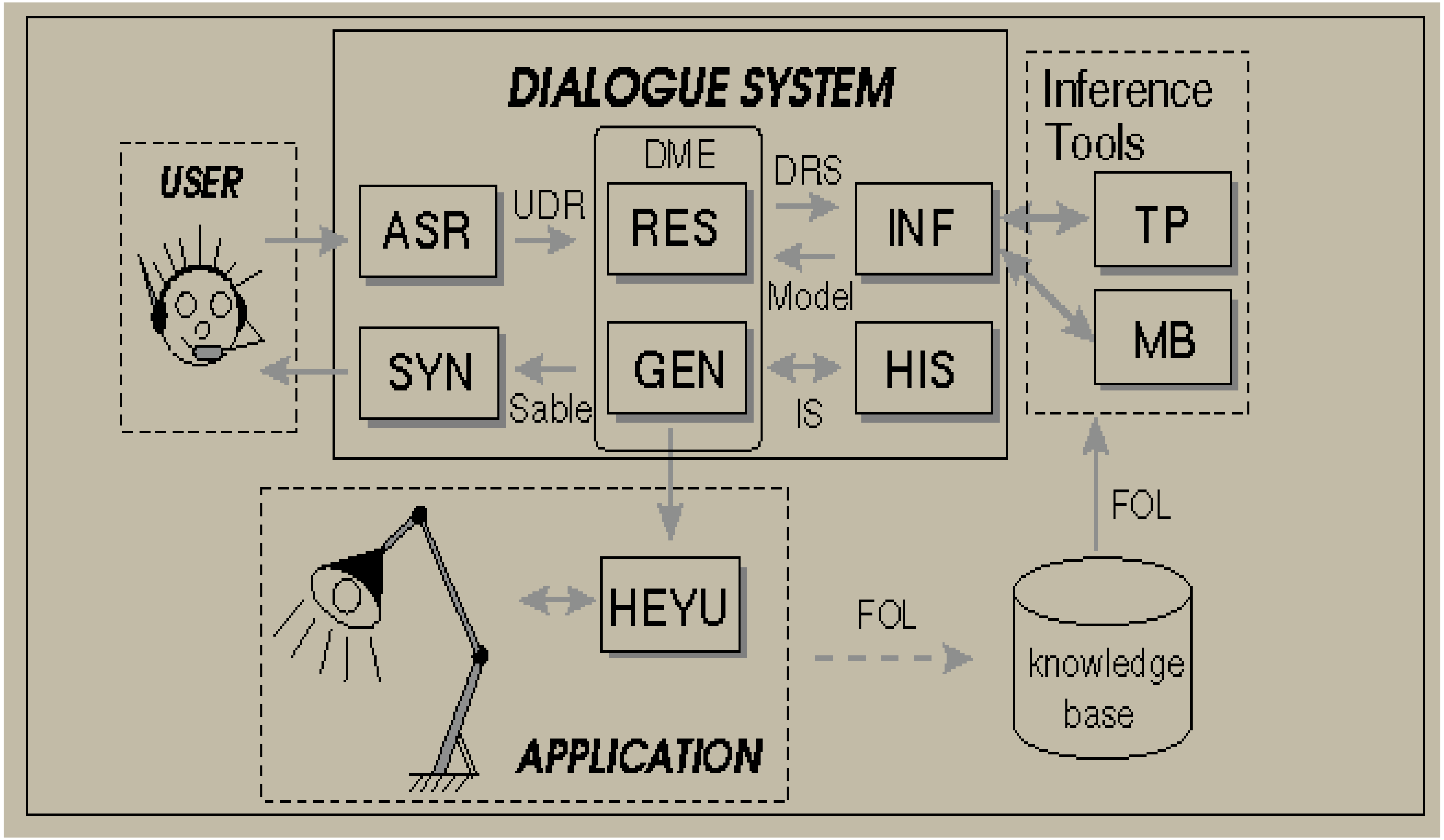
Part III: Applications

- Spoken Dialogue Systems with Embodied Agents
- Small domains, therefore feasible
- Two Applications:
  - Home Automation
  - Godot the Robot
- Example of the Beauty of Models
- Conclusions

Application 1: Home Automation

- Implemented as society of OAA agents:
  - ASR (speech recognition): NUANCE
  - SYN (synthesis): FESTIVAL
  - RES (resolution): DORIS
  - INF (inference): SPASS, MACE
- XML configuration of domain knowledge
- Application: Home Automation
  - X-10 and HEYU
  - Lights and Radio in ‘Smart Office’

Architecture



Heyu X-10
Home Configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE heyu_config SYSTEM "heyu_config.dtd">
<heyu>
 <use device="a1"/>
 <use device="a2"/>
 <use device="a3"/>
 <use device="a4"/>
 <label device="a1">lambda(P,merge(drs([X],[black(X),light(X)]),apply(P,X)))</label>
 <label device="a2">lambda(P,merge(drs([X],[red(X),light(X)]),apply(P,X)))</label>
 <label device="a3">lambda(P,merge(drs([X],[blue(X),light(X)]),apply(P,X)))</label>
 <label device="a4">lambda(P,merge(drs([X],[radio(X)]),apply(P,X)))</label>
</heyu>
```

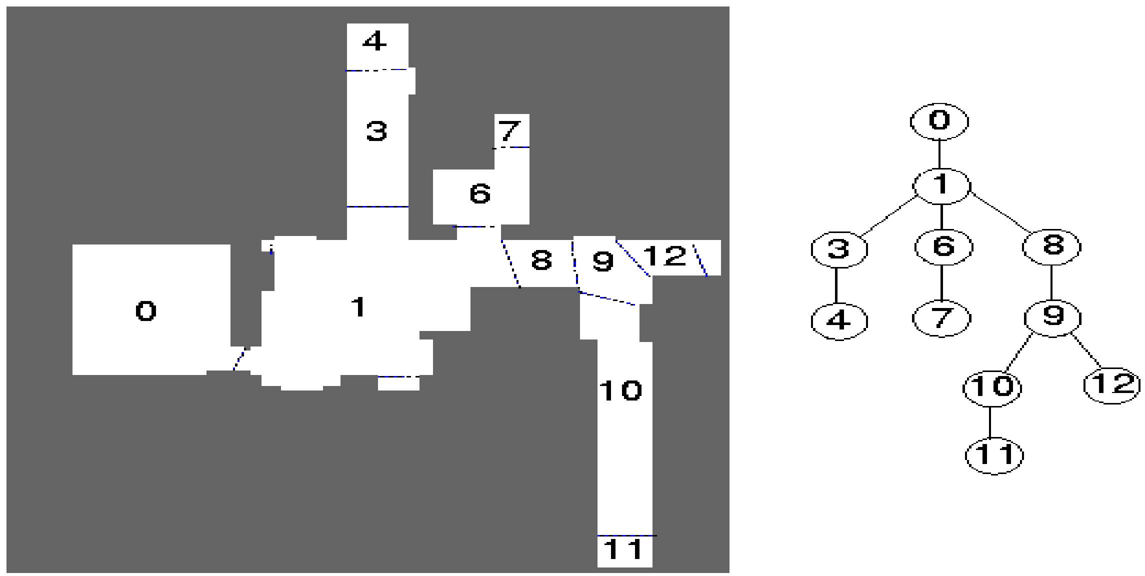
Application 2: Godot

- RWI Magellan Pro mobile robot platform
- Onboard PC running Linux
- Connected via wireless LAN
- Sensors:
  – 16 sonar (occupied space)
  – 16 infrared (distance)
  – 16 collision detectors
- CCD camera on pan-tilt unit
- Shaft encoders (odometry)

The Internal Map (1/2)

- Godot moves about in the basement of our department
- Internal map with two layers
  - geometrical layer: occupancy grid to represent occupied and free space
  - topological layer: automatically constructed using Voronoi diagram decomposition
- Semantic labels attached to regions of topological layer

The Internal Map (2/2)



- Numbers in the map are identifiers of topological regions
- Use these to associate semantic representations with regions

The Map Viewer

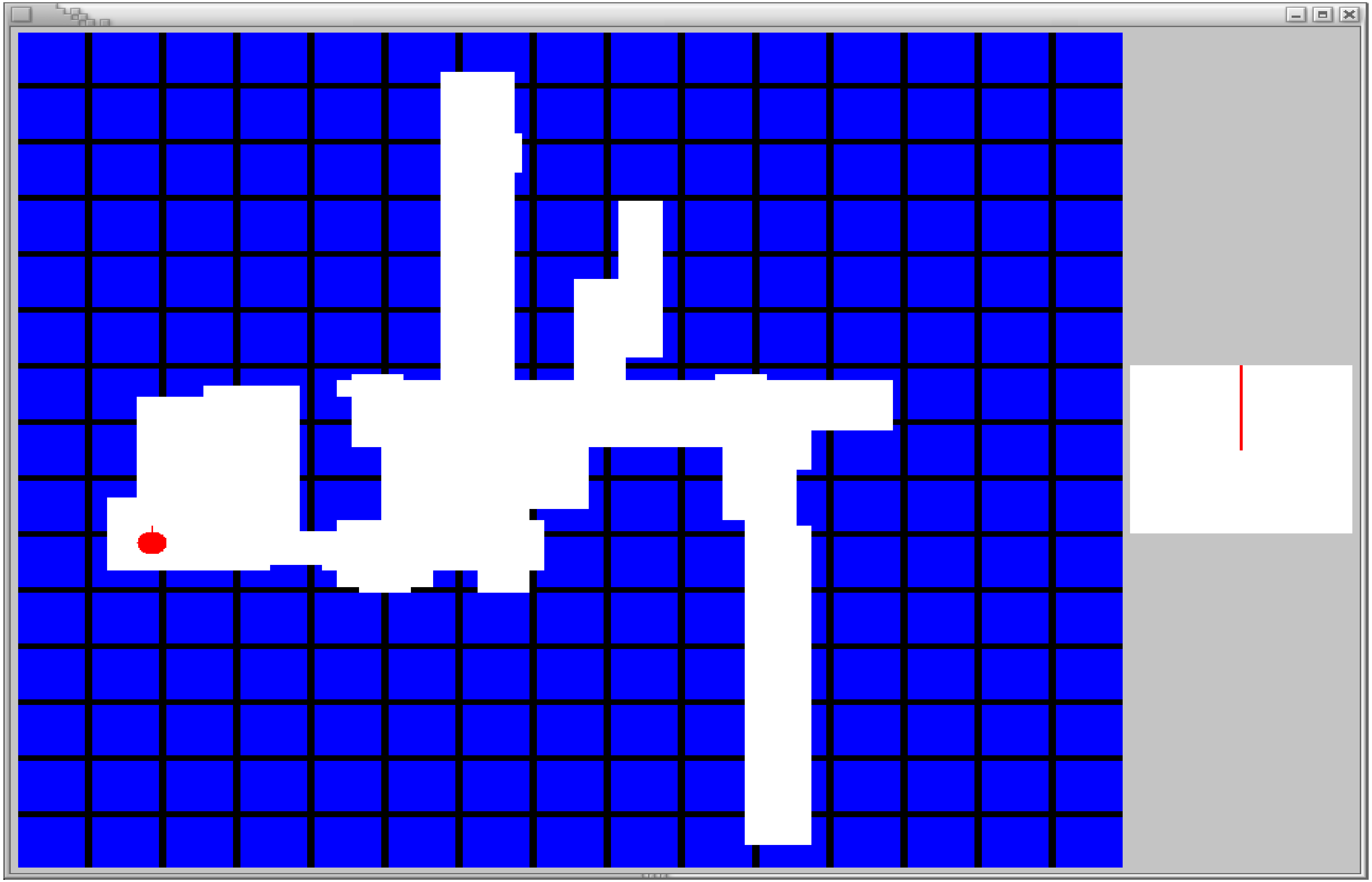
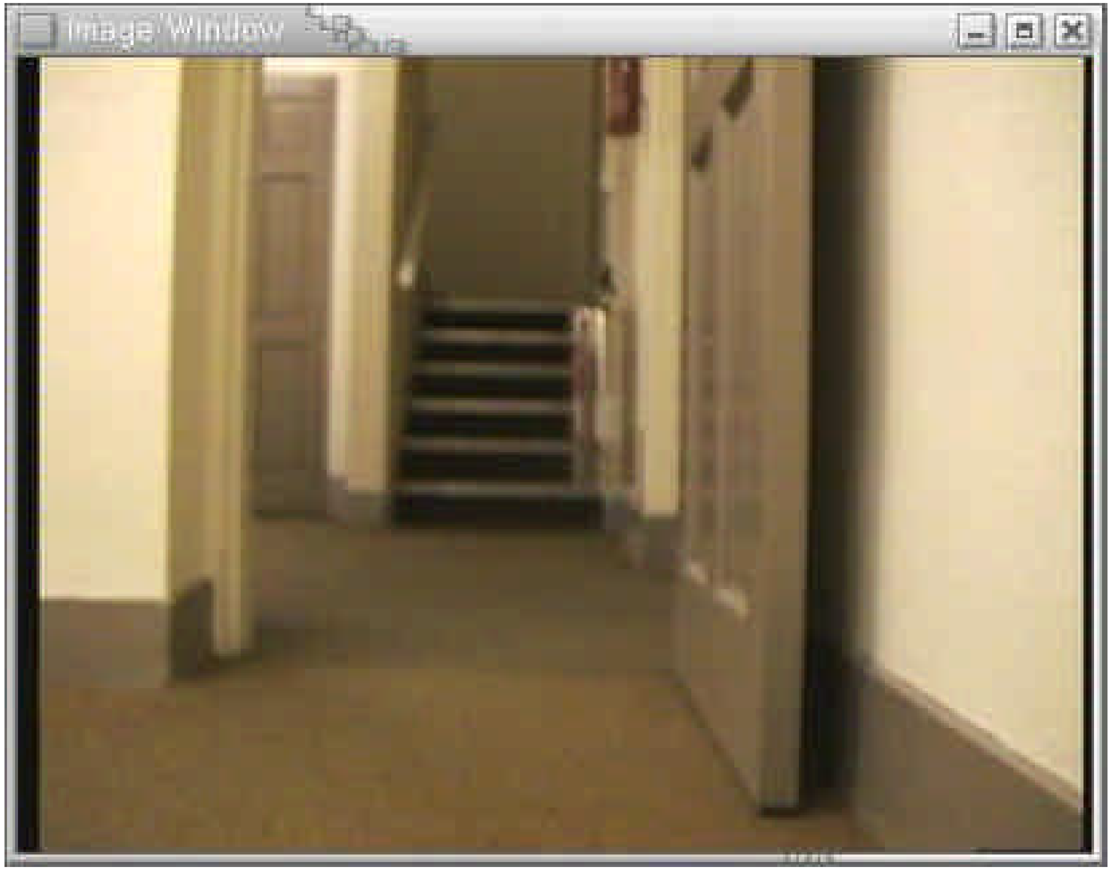


Image Viewer



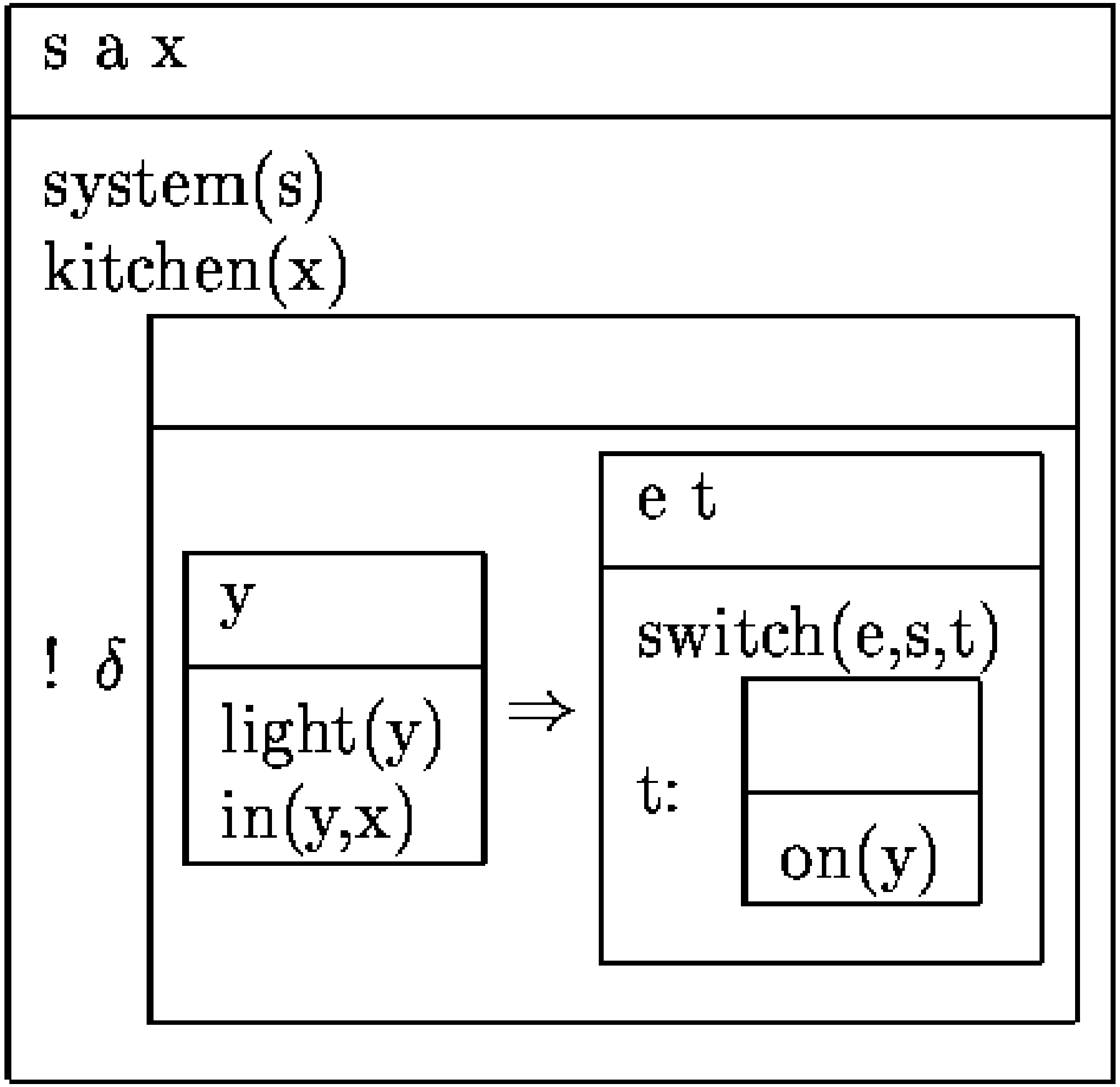
Example:
Dialogue with Godot

Camera
View

Beauty of Models
Example: Quantification

- Example Instructions:
  - "Clean all the rooms on the first floor!" (Robot)
  - "Turn of every light except the light in the kitchen!" (Home Automation)
- Model builder will produce a model with the number of primitives satisfied by the domain of quantification

Example DRS



- DRS for ‘Switch every light in the kitchen on’

Example: First-Order Model

- Instruction: “Switch every light in the kitchen on!”
  - $\exists$w $\exists$s $\exists$x(possible-world(x) & system(w,s) & kitchen(w,x) & $\exists$v$\exists$a (action(w,a,v) & $\forall$y (light(a,y) & in(a,y,x) $\rightarrow$ $\exists$e$\exists$t(switch(w,e,s,t) & on(t,y))))))
  - Output model:

```
D={d1,d2,d3,d4,d5,d6,d7,d8}
F(possible_world)={d1,d2,d3}
F(system)={(d1,d4),(d2,d4),(d3,d4)}
F(kitchen)={(d1,d5),(d2,d5),(d3,d5)}
F(action)={(d1,d2,d3)}
F(light)={(d1,d6),(d2,d6),(d3,d6),(d1,d7),(d2,d7),(d3,d7)}
F(in)={(d1,d6,d5),(d2,d6,d5),(d3,d6,d5),(d1,d7,d5),(d2,d7,d5),(d3,d7,d5)}
F(poweron={(d2,d6),(d2,d7)}
F(off)={(d1,d6),(d1,d7)}
F(on)={(d3,d6),(d3,d7)}
```

Summary

- Tools for computational semantics
  - theorem proving
  - model building
  - model checking
- Use first-order logic
- Semantic representations from speech input
  - Discourse Representation Theory
  - Ambiguity resolution
- Applications to spoken dialogue systems
- Role of model building

Conclusion: Inference in Dialogue Systems

- Slot-filling or frame-based approaches are old-fashioned
- Model building provides an alternative, opening a wide variety of interpretation tasks
- Based on formal theory of discourse
- Three reasons for first-order theorem proving to play a role in future systems
  - Theorem proving is still a promising emerging field
  - Not tuned to linguistic problems
  - Current approach is non-incremental