ECE 566

Knowledge Systems with Frames (structured objects)

G1
Element-of: Giving_events
giver: John
recip: Mary
obj: Book

OC2
Element-of: Occupation_events
worker: Mary
profession: Lawyer

OC1
Element-of: Occupation_events
worker: John
profession: Programmer

ADR1
Element-of: Address_events
person: John
location: 37-maple-st

JOHN
Element-of: Persons

MARY
Element-of: Persons

PROGRAMMER
Element-of: Jobs

LAWYER
Element-of: Jobs

37-MAPLE-ST
Element-of: Addresses

BOOK
Element-of: Phys_objs

PERSONS
Subset-of: Animals

Delineations and Prototypes

This type of deduction is used when an implication asserts properties about every member of a set.

Ex: Given the assertions "All Computer Science students have graduate standing", and "John is a Computer Science student", we should be able to deduce "John has graduate standing".

In predicate logic, we would have:

Fact: EL (John, CS_Students)

Rule: EL (?x, CS_Students) => EQ [class(?x), Grad]

Goal: EQ [class (John), Grad]

A production system may then go either backward or forward to prove the goal.

In frame language, we would have:

Fact:

John

Element-of: CS_Students

Goal:

John

class: Grad

In the frame formalism, a special kind of frame called a delineation unit is used to describe properties of each of the individuals in a set.

We use a delineation to describe Computer Science students:

?x | CS_Student
major: CS
class: Grad

Thus a delineation is a universal variable, whose domain of quantification is the given set.

Delineations are sometimes called Prototypes by some AI systems.

Delineation frames can then be used to generate new frames, by instantiating (substituting a constant for a variable) the delineation frame.

Ex:

John

Element-of: CS_Students

matches ?x | CS_Student (the variable ?x matches with element-of CS_Students). Applying the delineation unit, adds to the John fact unit the slots for major and class.

John

Element-of: CS_Students

major: CS

class: Grad

Goal with Existential Variables

A goal frame such as:

?y

class: Grad

is used to find out which individual has graduate standing.

In the backward direction, matching this with frame
?x | CS_Students would create the subgoal

?y

Element-of: CS_Students

This subgoal then matches the John unit, with substitution {John/?y}.

Procedural Attachment

We can associate computer programs with slots of delineations. Then these programs can be invoked to execute (with argument values) to produce slot values for instances of the delineation.

Ex: Suppose a given slot-value is equal to the multiplication of two other slot-values (e.g. V=IR).

?x | Resistor
I: (current through R)
R: (resistance value)
V:Times[I(?x), R(?x)]

Procedures occurring in delineations are executed whenever the value of the given slot is accessed.

Frame Rules

Are used for expressing information not about all members of a set. Frame rules have two parts, ANTE and CONSE. Each is a list of frame units possibly containing variables.

In the forward direction, if all ANTE frames are matched, instantiated CONSE frames are added to the set of fact frames. If some of the fact frames (slots, slot values) already exist, we only add the new properties mentioned in the CONSE frames.

In the backward direction, one of the CONSE frames must match the goal frame. If this succeeds, the frames in the ANTE are set up as subgoal frames.

Ex: Suppose we want to represent that if a department has a manager and some worker, then they both work in the same department and the manager is the boss of the other person.

R1

ANTE:

?x
  Element-of: Departments
  manager: ?y

?z
  Element-of: Employees
  works_in: ?x

?y
  Element-of: Employees

CONSE:

?y
  Element-of: Employees
  works_in: ?x
  boss_of: ?z

?z
  Element-of: Employees
  works_in: ?x
  boss: ?y

Review and Exercises

1. Represent the following information:

   "John gave Mary the C Book."

$$
\begin{aligned}
(\exists\, ?z)[\ & \mathrm{EL}(?z, \mathrm{Giving\_Events}) \wedge \\
& \mathrm{EQ}(\mathrm{giver}(?z), \mathrm{John}) \wedge \\
& \mathrm{EQ}(\mathrm{recip}(?z), \mathrm{Mary}) \wedge \\
& \mathrm{EQ}(\mathrm{obj}(?z), \mathrm{C\_book})]
\end{aligned}
$$

↓ Skolemize ?z

G1

Element-of: Giving_Events

giver: John

recip: Mary

obj: C_book

2. Show how the following query is represented and answered using frameds:

"What did John give Mary?"

First we construct a Goal frame. Recall that in goal units variables have existential, while in fact units they are assumed to have universal quantification.

?x

EL: Giving_Events

giver: John

recip: Mary

obj: ?v

Assumed Existential

S = {G1/?x, C_book/?v}

3. Show how the following information is represented.

"John gave something to everyone."

<u>In first order logic:</u>

EL(?x, ?y)	←	?x is an element of set ?y
Giving_Events	←	set name, constant
John	←	constant to represent John
giver(?x)	←	giver in the giving event ?x
recip(?x)	←	recipient in the giving event ?x
obj(?x)	←	what is given in event ?x

$$(\forall\ ?y)(\exists\ ?z)(\exists\ ?t)[EL(?z, Giving\_Events) \wedge$$
$$EQ(giver(?z), John) \wedge$$
$$EQ(recip(?z), ?y) \wedge$$
$$EQ(obj(?z), ?t)]$$

Skolemizing existential variables, we get:

$$(\forall ?y)[\, EL(g(?y), Giving\_Events) \wedge$$
$$EQ(giver(g(?y)), John) \wedge$$
$$EQ(recip(g(?y)), ?y) \wedge$$
$$EQ(obj(g(?y)), sk(?y))]$$

g(?y)

Element-of: Giving_Events

giver: John

recip: ?y

obj: sk(?y)

S2 = {Mary/?y, sk(Mary)/?v, g(Mary)/?x}

4. Suppose we have two frame types, Dept frames and Employee frames:

Dept
- Element-of: departments
- name: dept. name
- manager: an employee

Employee
- Element-of: employees
- name: employee name
- works-in: a department
- boss: an employee

(a)Write the following expressions as frames

Manager(PURCHASING-DEPT, JOHN-JONES)
Works-in(PURCHASING-DEPT, JOE-SMITH)

Manager(PURCHASING-DEPT, JOHN-JONES)
Works-in(PURCHASING-DEPT, JOE-SMITH)

e1

Element-of: employees
name: JOE-SMITH
works-in: d1

d1

Element-of: departments
name: PURCHASING-DEPT
manager: e2

e2

Element-of: employees
name: JOHN-JONES

(b) Write the following as a frame-based rule.

$$[\text{Works-in}(?x, ?y) \wedge \text{Manager}(?z, ?y)] \Rightarrow \text{Boss-of}(?y, ?z)$$

ANTE:

?y
Element-of: employees
works-in: ?x

?x
Element-of: departments
manager: ?z

?z
Element-of: employees

CONSE:

?y
Element-of: employees
works-in: ?x
boss: ?z

?z
Element-of: employees
works-in: ?x

(c) Using your frame system, show that:

JOHN-JONES is the boss of JOE-SMITH

e1

Element-of: employees

name: JOE-SMITH

works-in: d1

d1

Element-of: departments

name: PURCHASING-DEPT

manager: e2

e2

Element-of: employees

name: JOHN-JONES