

CarPool World in SNePSLOG

An Example of Formalizing a Domain

CSE 663, Advanced Knowledge Representation, Fall 2004

Stuart C. Shapiro
Department of Computer Science and Engineering
and Center for Cognitive Science
201 Bell Hall
University at Buffalo, The State University of New York
Buffalo, NY 14260-2000
shapiro@cse.buffalo.edu

September 1, 2004

1 The Domain

To formalize a domain, we need to

1. define the atomic symbols we will use;
2. state and formalize the domain rules;
3. state and formalize the ground facts.

Then we can test the formalization with some questions about the situation.

I will give an example of this process using CarPool World, which is a small domain consisting of the people Tom, Betty, and their mothers. Tom is male, Betty is female, and, of course, anyone's mother is female. On any given day, one person may drive one or more others in a car, in which case the former is the driver and the others are passengers. No one can be both a driver and a passenger. On the day we are interested in, Tom's mother drives Tom, and Betty drives her mother. The questions we are interested in answering are:

1. Who are the drivers?
2. Who are the passengers?
3. Is someone both a driver and a passenger?
4. If Betty drove Tom, would she be the driver?
5. If Betty drove Tom, would Betty's mother be a passenger?
6. Is a female driving a male?
7. Who is a male who is not a driver?

2 The Formalization

I will now present the formalization, separated into the categories listed above.

ATOMIC SYMBOLS

Individual Constants

Betty: The individual named Betty.

Tom: The individual named Tom.

Larry: The individual named Larry. (Added to provide someone who's not involved in a car pool.)

Function Symbols

Individual-Valued Function Symbols

motherOf(x): The mother of x .

Proposition-Valued Function Symbols

Driver(x): The proposition that x is the driver of a car.

Female(x): The proposition that x is female.

Male(x): The proposition that x is male.

Passenger(x): The proposition that x is a passenger in a car.

Person(x): The proposition that x is a person.

Drives(x,y): The proposition that x drives y in a car.

DOMAIN RULES (NONLOGICAL AXIOMS)

Everyone is male or female, but not both.

$all(x)(Person(x) \Rightarrow andor(1, 1)\{Male(x), Female(x)\})$.

Everyone's mother is female.

$all(x)(Person(x) \Rightarrow Female(motherOf(x)))$.

If someone drives someone in a car, the former is the driver and the latter is a passenger.

$all(x, y)(Drives(x, y) \Rightarrow \{Driver(x), Passenger(y)\})$.

No one can be both a driver and a passenger.

$all(x)(Person(x) \Rightarrow andor(0, 1)\{Driver(x), Passenger(x)\})$.

At most one person drives any other person.

$all(p)(Person(p) \Rightarrow nexists(-, 1, -)(x)(Person(x) : Drives(x, p)))$.

GROUND FACTS

Tom, Betty, and their mothers are people.

$Person(\{Tom, Betty, motherOf(Tom), motherOf(Betty)\})$.

Larry is a person.

$Person(Larry)$

Tom is male and Betty is female.

$Male(Tom) \text{ and } Female(Betty)$.

Tom's mother drives Tom.

$Drives(motherOf(Tom), Tom)$.

Betty drives her mother.

$Drives(Betty, motherOf(Betty))$.

QUESTIONS

Who are the drivers?

$Driver(?x)$?

List the passengers.

$askwh Passenger(?x)$

Is someone both a driver and a passenger?
Driver(?x) and Passenger(?x)?

If Betty drove Tom, would she be the driver?
Drives(Betty, Tom) => Driver(Betty)?

If Betty drove Tom, would Betty's mother be a passenger?
Drives(Betty, Tom) => Passenger(motherOf(Betty))?

Is a female driving a male?
ask andor(3, 3)Drives(?x, ?y), Female(?x), Male(?y)

Who is a male who is not a driver?
askwh andor(2, 2)Male(?x), Driver(?x)

Is Larry driving Tom?
Drives(Larry, Tom)?

3 The Run

Finally, I will show a run of SNePSLOG in which the above knowledge base is built, and the questions are asked and answered. I will start by loading SNePS into Common Lisp. The following is edited only to adjust line and page breaks so that it is easy to read.

```
cl-user(1): :ld /projects/snwiz/bin/sneps
; Loading /projects/snwiz/bin/sneps.lisp
Loading system SNePS...10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
SNePS-2.6 [PL:1a 2004/08/26 23:05:27] loaded.
Type '(sneps)' or '(snepslog)' to get started.

cl-user(2): (snepslog)

Welcome to SNePSLOG (A logic interface to SNePS)

Copyright (C) 1984--2004 by Research Foundation of
State University of New York. SNePS comes with ABSOLUTELY NO WARRANTY!
Type 'copyright' for detailed copyright information.
Type 'demo' for a list of example applications.

: demo "carPoolWorld.snepslog"

File /projects/shapiro/CSE663/carPoolWorld.snepslog is now the source of input.

: ;;; CarPool World in SNePSLOG
;;; an example of formalizing a domain
;;; Stuart C. Shapiro
;;; August 30, 2004
;;;
;;; ATOMIC SYMBOLS
;;;
;;; Individual Constants
;;; =====
;;; Betty: The individual named Betty.
;;; Tom: The individual named Tom.
;;; Larry: The individual named Larry.
;;; (Added to provide someone who's not involved in a car pool.)
```

```

;;; Function Symbols
;;; =====
;;;
;;; Individual-Valued Function Symbols
;;; -----
;;; motherOf(x): The mother of x.
;;;
;;; Proposition-Valued Function Symbols
;;; -----
;;; Driver(x): The proposition that x is the driver of a car.
;;; Female(x): x is female.
;;; Male(x): x is male.
;;; Passenger(x): The proposition that x is a passenger in a car.
;;; Person(x): x is a person.
;;;
;;; Drives(x,y): The proposition that x drives y in a car.

;;; Start with a fresh knowledge base
clearkb
Knowledge Base Cleared

:
;;; Turn tracing off (on)
^(setf *infertrace* nil)
nil

: ;;;^(setf *infertrace* :surface)

;;; Domain Rules (Nonlogical Axioms)
;;;=====

;;; Everyone is male or female, but not both.
all(x)(Person(x) => andor(1,1){Male(x), Female(x)}).

    all(x)(Person(x) => (andor(1,1){Female(x),Male(x)}))

CPU time : 0.00

:
;;; Everyone's mother is female.
all(x)(Person(x) => Female(motherOf(x))).

    all(x)(Person(x) => Female(motherOf(x)))

CPU time : 0.00

:
;;; If someone drives someone in a car,
;;;   the former is the driver and the latter is a passenger.
all(x,y)(Person({x,y}) => (Drives(x,y) => {Driver(x), Passenger(y)})).

    all(y,x)(Person({y,x}) => ({Drives(x,y)} v=> {Passenger(y),Driver(x)}))

```

```

CPU time : 0.00

:
;;; No one can be both a driver and a passenger.
all(x)(Person(x) => andor(0,1){Driver(x), Passenger(x)}).

    all(x)(Person(x) => (andor(0,1){Passenger(x),Driver(x)}))

CPU time : 0.00

:
;;; At most one person drives any other person.
all(p)(Person(p) => nexists(_,1,_)(x)({Person(x)}: {Drives(x,p)})).

    all(p)(Person(p) => (nexists(_,1,_)(x)[{Person(x)}:{Drives(x,p)}]))

CPU time : 0.00

:
;;; Ground Facts
;;;=====

;;; Tom, Betty, and their mothers are people.
Person({Tom, Betty, motherOf(Tom), motherOf(Betty)}).

    Person({motherOf(Betty),motherOf(Tom),Betty,Tom})

CPU time : 0.00

:
;;; Larry is a person.
Person(Larry).

    Person(Larry)

CPU time : 0.00

:
;;; Tom is male and Betty is female.
Male(Tom) and Female(Betty).

    Female(Betty) and Male(Tom)

CPU time : 0.00

:
;;; Tom's mother drives Tom.
Drives(motherOf(Tom), Tom).

    Drives(motherOf(Tom),Tom)

CPU time : 0.00

```

```

:
;;; Betty drives her mother.
Drives(Betty, motherOf(Betty)).

    Drives(Betty,motherOf(Betty))

CPU time : 0.00

:
;;; Questions
;;;=====

;;; Who are the drivers?
Driver(?x)?

    ~Driver(motherOf(Betty))
    ~Driver(Tom)
    Driver(Betty)
    Driver(motherOf(Tom))

CPU time : 0.31

:
;;; List the passengers.
askwh Passenger(?x)
    motherOf(Betty)
    Tom

:
;;; Is someone both a driver and a passenger?
Driver(?x) and Passenger(?x)?

    andor(0,1){Driver(Tom),Passenger(Tom)}
    andor(0,1){Driver(motherOf(Betty)),Passenger(motherOf(Betty))}
    andor(0,1){Driver(Betty),Passenger(Betty)}
    andor(0,1){Driver(motherOf(Tom)),Passenger(motherOf(Tom))}

CPU time : 0.07

:
;;; If Betty drove Tom, would she be the driver?
Drives(Betty, Tom) => Driver(Betty)?

    Drives(Betty,Tom) => Driver(Betty)

CPU time : 0.01

:
;;; If Betty drove Tom, would Betty's mother be a passenger?
Drives(Betty, Tom) => Passenger(motherOf(Betty))?

CPU time : 0.00

```

```
:
;;; Is a female driving a male?
ask andor(3,3){Drives(?x, ?y), Female(?x), Male(?y)}
  Female(motherOf(Tom)) and Drives(motherOf(Tom),Tom) and Male(Tom)

:
;;; Who's a male who is not a driver?
askwh andor(2,2){Male(?x), ~Driver(?x)}
  Tom

:
;;; Is Larry driving Tom?
Drives(Larry, Tom)?

  ~Drives(Larry, Tom)

CPU time : 0.00

:

End of /projects/shapiro/CSE663/carPoolWorld.snepslog demonstration.

CPU time : 1.07
```