Knowledge Representation and Reasoning

Logics for Artificial Intelligence

Stuart C. Shapiro

Department of Computer Science and Engineering
and Center for Cognitive Science
University at Buffalo, The State University of New York
Buffalo, NY 14260-2000

shapiro@cse.buffalo.edu

Contents

8 SNePS

SNePSLOG Syntax
Atomic Symbols

Individual Constants, Variables,
Function Symbols, Predicate Symbols:

any Lisp symbol, number, or string.

All that matters is the sequence of characters.
I.e. `"4"`, `\4`, and `4`, are the same.

The sets of individual constants, variables, function symbols, and predicate symbols should be distinct, but don't have to be.

SNePSLOG Syntax
Terms

An individual constant is a term.

A variable is a term.

If `t1, ..., tn` are terms, then {`t1, ..., tn`} is a set of terms.

If `t1, ..., tn` are terms or sets of terms and `f` is a function symbol, then `f(t1, ..., tn)` is a term.

A function symbol needn't have a fixed arity, but it might be a mistake of formalization otherwise.

SNePSLOG Syntax
Wffs 1

If `t1, ..., tn` are terms or sets of terms and `P` is a predicate symbol, then `P(t1, ..., tn)` is a wff.

A predicate symbol needn't have a fixed arity, but it might be a mistake of formalization otherwise.

SNePSLOG Syntax

Wffs 2

If `P` is a set of wffs
and `i` and `j` are integers such that `0 <= i <= j`, then

`andor``(i, j)P` is a wff, and

`thresh``(i, j)P` is a wff.

If `P` and `Q` are sets of wffs, then

`P &=> Q` is a wff, and

and `P v=> Q` is a wff.

SNePSLOG Syntax
Wffs 3

If `P` is a wff and `x1, ..., xn` are variables, then

`all(x1, ..., xn)(P)` is a wff.

If `P` and `Q` are sets of wffs, `x1, ..., xn` are variables, and `i`, `j`, and `k` are integers such that `0 <= i <= j <= k`, then

`nexists(i, j, k)(x1, ..., xn)(P: Q)` is a wff.

SNePSLOG Syntax

Summary of Nonatomic Wffs

```
andor(i, j){P1, ..., Pn}

thresh(i, j){P1, ..., Pn}

{P1, ..., Pn} &=> {Q1, ..., Qm}

{P1, ..., Pn} v=> {Q1, ..., Qm}

all(x1, ..., xn)(P)

nexists(i, j, k)(x1, ..., xn)({P1, ..., Pn}: {Q1, ..., Qm})
```

SNePSLOG Syntax
Abbreviations

`~P` is an abbreviation of `andor(0,0){P}`
`P and Q` is an abbreviation of `andor(2,2) {P,Q}`
`P or Q` is an abbreviation of `andor(1,2){P, Q}`

`P <=> Q` is an abbreviation of `thresh(1,1){P, Q}`

`P1 <=>` ··· `<=> Pn`
is an abbreviation of `thresh(1, n-1){P1, ..., Pn}`

`thresh(i){P1, ..., Pn}`
is an abbreviation of `thresh(i, n-1){P1, ..., Pn}`

`{P1, ..., Pn} => {Q1, ..., Qm}`
is an abbreviation of `{P1, ..., Pn} v=> {Q1, ..., Qm}`

SNePSLOG Syntax
More Abbreviations

`nexists(_, j, _)(x1, ..., xn)(P: Q)` is an abbreviation of `nexists(0, j, inf)(x1, ..., xn)(P: Q)`

`nexists(i, _, k)(x1, ..., xn)(P: Q)` is an abbreviation of `nexists(i, inf, k)(x1, ..., xn)(P: Q)`

SNePSLOG Syntax
Wffs are Terms

Every wff is a term.

So, e.g., `Believes(Tom, ~Penguin(Tweety))`
is a wff, and a well-formed term.

SNePSLOG Semantics The Intensional Domain of (Mental) Entities

Frege: *The Morning Star is the Evening Star.*
different from *The Morning Star is the Morning Star.*

Russell: *George IV wanted to know whether Scott was the author of Waverly.*
not *George IV wanted to know whether the author of Waverly was the author of Waverly.*

Intensions vs. Extensions

Morning Star and *Evening Star*
Scott and *the author of Waverly*

are different intensions, or intensional entities, or mental entities, or just entities,

even though they are coreferential, or extensionally equivalent, or have the same extensions.

SNePSLOG Semantics
Intensional Representation

SNePSLOG individual constants and ground terms denote intensions, (mental) entities.

SNePSLOG Semantics
Reified Propositions

Mental entities include propositions.

Propositions are first-class members of the domain.

SNePSLOG wffs denote propositions.

SNePSLOG Semantics
AndOr and Thresh

$[\![$`andor(i, j)`$\{$`P1, ..., Pn`$\}]\!]$ = The proposition that at least `i` and at most `j` of $[\![$P1$]\!]$, . . . , $[\![$Pn$]\!]$ are true.

$[\![$`thresh(i, j)`$\{$`P1, ..., Pn`$\}]\!]$ = The proposition that either fewer than `i` or more than `j` of $[\![$P1$]\!]$, . . . , $[\![$Pn$]\!]$ are true.

SNePSLOG Semantics
And Entailment and Or Entailment

⟦`{P1, ..., Pn} &=> {Q1, ..., Qm}`⟧
= The proposition that if ⟦P1⟧ and . . . and ⟦Pn⟧ are true, then so is ⟦Qi⟧, `1 <= i <= m`.

⟦`{P1, ..., Pn} v=> {Q1, ..., Qm}`⟧
= The proposition that if any ⟦Pi⟧, for `1 <= i <= n`, is true, then so is ⟦Qj⟧, `1 <= j <= m`.

SNePSLOG Semantics Universally and Numerically Quantified Wffs

$[\![\texttt{all(x1, ..., xn)(P)}]\!]$
$=$ The proposition that $[\![\texttt{P}\{\texttt{t1/x1, ..., tn/xn}\}]\!]$,
for every sequence of ground terms $\texttt{t1, ..., tn}$, $\texttt{ti} \neq \texttt{tj}$,
is true.

$[\![\texttt{nexists(i, j, k)(x1, ..., xl)(}\{\texttt{P1, ..., Pn}\}\texttt{:}$
$\{\texttt{Q1, ..., Qm}\}\texttt{)}]\!] =$
There are $\texttt{k}$ sequences of distinct ground terms $\texttt{t1,...,tl}$,
and, of them, at least $\texttt{i}$, and at most $\texttt{j}$ satisfy
$[\![\texttt{Q1}\{\texttt{t1/x1, ..., tl/xl}\}]\!]$ and, . . . , and $[\![\texttt{Qm}\{\texttt{t1/x1, ..., tl/xl}\}]\!]$.

SNePSLOG Proof Theory
The Knowledge Base

Think of the SNePS KB as the contents of the mind of an intelligent agent.

The terms in the KB denote mental entities that the agent has conceived of (so far).

Some of the wffs are **asserted**.
These denote propositions that the agent believes.

The rules of inference sanction believing some additional proposition(s), but drawing that inference is optional.
I.e., the agent is not logically omniscient.

SNePSLOG Proof Theory
Implemented Rules of Inference
Reduction Inference

Reduction Inference$_1$: If α is a set of terms and $\beta \subset \alpha$,
$\mathtt{P(t}_1\mathtt{,\ldots,}\alpha\mathtt{,\ \ldots t}_n\mathtt{)} \vdash \mathtt{P(t}_1\mathtt{,\ldots,}\beta\mathtt{,\ldots\ t}_n\mathtt{)}$

Reduction Inference$_2$: If α is a set of terms, and $\mathtt{t} \in \alpha$,
$\mathtt{P(t}_1\mathtt{,\ldots,}\alpha\mathtt{,\ \ldots t}_n\mathtt{)} \vdash \mathtt{P(t}_1\mathtt{,\ \ldots,\ t,\ \ldots t}_n\mathtt{)}$

Example of Reduction Inference

```
: clearkb
Knowledge Base Cleared

: Member({Fido, Rover, Lassie}, {dog, pet}).

  Member({Lassie,Rover,Fido},{pet,dog})

 CPU time : 0.01

: Member ({Fido, Lassie}, dog)?

  Member({Lassie,Fido},dog)

 CPU time : 0.00
```

SNePSLOG Proof Theory Implemented Rules of Inference for AndOr

AndOr $\mathbf{I}_1$: $\mathtt{P}_1$, ..., $\mathtt{P}_n \vdash$ $\mathtt{andor}(n,n)\{\mathtt{P}_1, \ldots, \mathtt{P}_n\}$

AndOr $\mathbf{I}_2$: $\sim\mathtt{P}_1$, ..., $\sim\mathtt{P}_n \vdash$ $\mathtt{andor(0,0)}\{\mathtt{P}_1, \ldots, \mathtt{P}_n\}$

AndOr $\mathbf{E}_1$: $\mathtt{andor}(i,j)\{\mathtt{P}_1, \ldots, \mathtt{P}_n\}$, $\sim\mathtt{P}_1, \ldots, \sim\mathtt{P}_{n-i} \vdash \mathtt{P}_j$
for $n-i < j \leq n$

AndOr $\mathbf{E}_2$: $\mathtt{andor}(i,j)\{\mathtt{P}_1, \ldots, \mathtt{P}_n\}$, $\mathtt{P}_1, \ldots, \mathtt{P}_j \vdash \sim\mathtt{P}_k$,
for $j < k \leq n$

SNePSLOG Proof Theory Implemented Rules of Inference for Thresh

Thresh $\mathbf{E}_1$: When at least i args are true, and at least $n-j-1$ args are false, conclude that any other arg is true.
$\mathtt{thresh}(i,j)\{\mathtt{P}_1,\ \ldots,\ \mathtt{P}_n\}$,
$\mathtt{P}_1,\ \ldots,\ \mathtt{P}_i,\ \neg\mathtt{P}_{i+1},\ \ldots,\ \neg\mathtt{P}_{i+n-j-1} \vdash \mathtt{P}_{i+n-j}$

Thresh $\mathbf{E}_2$: When at least $i-1$ args are true, and at least $n-j$ args are false, conclude that any other arg is false.
$\mathtt{thresh}(i,j)\{\mathtt{P}_1,\ \ldots,\ \mathtt{P}_n\}$,
$\mathtt{P}_1,\ \ldots,\ \mathtt{P}_{i-1},\ \neg\mathtt{P}_{i+1},\ \ldots,\ \neg\mathtt{P}_{i+n-j} \vdash \neg\mathtt{P}_i$

SNePSLOG Proof Theory Implemented Rules of Inference for &=>

&=>I: If $\mathcal{A}$, $\mathtt{P}_1$, ..., $\mathtt{P}_n \vdash \mathtt{Q}_i$ for $1 \leq i \leq m$
then $\mathcal{A} \vdash \{\mathtt{P}_1$, ..., $\mathtt{P}_n\}$ &=> $\{\mathtt{Q}_1$, ..., $\mathtt{Q}_m\}$

&=>E: $\{\mathtt{P}_1$, ..., $\mathtt{P}_n\}$ &=> $\{\mathtt{Q}_1$, ..., $\mathtt{Q}_m\}$, $\mathtt{P}_1$, ..., $\mathtt{P}_n \vdash \mathtt{Q}_i$,
for $1 \leq i \leq m$

SNePSLOG Proof Theory
Implemented Rules of Inference
for `v=>`

`v=>I`: If $\mathcal{A}$, $\mathtt{P}_i \vdash \mathtt{Q}_j$ for $1 \leq i \leq n, 1 \leq j \leq m$
then $\mathcal{A} \vdash \{\mathtt{P}_1\texttt{, ..., }\mathtt{P}_n\}$ `v=>` $\{\mathtt{Q}_1\ \texttt{...}\mathtt{Q}_m\}$

`v=>E`: $\{\mathtt{P}_1\texttt{, ..., }\mathtt{P}_n\}$ `v=>` $\{\mathtt{Q}_1\texttt{, ..., }\mathtt{Q}_m\}$, $\mathtt{P}_i, \vdash \mathtt{Q}_j$,
for $1 \leq i \leq n, 1 \leq j \leq m$

SNePSLOG Proof Theory
Implemented Rules of Inference
for `all`

all E_1: $\texttt{all(x}_1\texttt{, ..., x}_l\texttt{)(\{P}_1\texttt{(x}_1\texttt{, ..., x}_l\texttt{), ..., P}_n\texttt{(x}_1\texttt{, ..., x}_l\texttt{)\} \&=> \{Q}_1\texttt{(x}_1\texttt{, ..., x}_l\texttt{), ..., Q}_m\texttt{(x}_1\texttt{, ..., x}_l\texttt{)\}}$, $\texttt{P}_1\texttt{(t}_1\texttt{, ..., t}_l\texttt{), ..., P}_n\texttt{(t}_1\texttt{, ..., t}_l\texttt{)} \vdash \texttt{Q}_i\texttt{(t}_1\texttt{, ..., t}_l\texttt{)}$, for $1 \leq i \leq m$

all E_2: $\texttt{all(x}_1\texttt{, ..., x}_l\texttt{)(\{P}_1\texttt{(x}_1\texttt{, ..., x}_l\texttt{), ..., P}_n\texttt{(x}_1\texttt{, ..., x}_l\texttt{)\} v=> \{Q}_1\texttt{(x}_1\texttt{, ..., x}_l\texttt{), ..., Q}_m\texttt{(x}_1\texttt{, ..., x}_l\texttt{)\}}$, $\texttt{P}_i\texttt{(t}_1\texttt{, ..., t}_l\texttt{)} \vdash \texttt{Q}_j\texttt{(t}_1\texttt{, ..., t}_l\texttt{)}$, for $1 \leq i \leq n, 1 \leq j \leq m$

UVBR

In any substitution $\{t_1/x_1, \ldots, t_n/x_n\}$, if $x_i \neq x_j$, then $t_i \neq t_j$

```
: all(u,v,x,y)(childOf({u,v}, {x,y}) => Siblings({u,v})).

: childOf({Tom,Betty,John,Mary}, {Pat,Harry}).

: Siblings({?x,?y})?

  Siblings({Mary,John})
  Siblings({John,Betty})
  Siblings({Betty,Tom})
  Siblings({Mary,Betty})
  Siblings({John,Tom})
  Siblings({Mary,Tom})
```

SNePSLOG Proof Theory
Implemented Rules of Inference
for `nexists`

nexists E$_1$: `nexists(i, j, k)`$(\overline{x})$`(`$\{$`P`$_1(\overline{x})$`, ..., P`$_n(\overline{x})\}$`:` $\{$`Q`$(\overline{x})\}$`),`
`P`$_1(\bar{t}_1)$`, ..., P`$_n(\bar{t}_1)$`, Q`$(\bar{t}_1)$`, ..., P`$_1(\bar{t}_j)$`, ..., P`$_n(\bar{t}_j)$`, Q`$(\bar{t}_j)$`,`
`P`$_1(\bar{t}_{j+1})$`, ..., P`$_n(\bar{t}_{j+1}) \vdash \neg$ `Q`$(\bar{t}_{j+1})$

nexists E$_2$: `nexists(i, j, k)`$(\overline{x})$`(`$\{$`P`$_1(\overline{x})$`, ..., P`$_n(\overline{x})\}$`:` $\{$`Q`$(\overline{x})\}$`),`
`P`$_1(\bar{t}_1)$`, ..., P`$_n(\bar{t}_1)$`,` $\neg$ `Q`$(\bar{t}_1)$`, ..., P`$_1(\bar{t}_{k-i})$`, ..., P`$_n(\bar{t}_{k-i})$`,`
$\neg$ `Q`$(\bar{t}_{k-i})$`, P`$_1(\bar{t}_{k-i+1})$`, ..., P`$_n(\bar{t}_{k-i+1}) \vdash$ `Q`$(\bar{t}_{k-i+1})$

Loading SNePSLOG

```
cl-user(1): :ld /projects/shapiro/Sneps/sneps262
...
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.
```

Running SNePSLOG

```
: clearkb
Knowledge Base Cleared

: Member({Fido, Rover, Lassie}, {dog, pet}).
  Member({Lassie,Rover,Fido},{pet,dog})
 CPU time : 0.01

: Member ({Fido, Lassie}, dog)?

I wonder if  Member({Lassie,Fido},dog)
holds within the BS defined by context default-defaultct

I know  Member({Lassie,Rover,Fido},{pet,dog})

  Member({Lassie,Fido},dog)
 CPU time : 0.01
```

Common SNePSLOG Commands

```
: clearkb
Knowledge Base Cleared

: all(x)(dog(x) => animal(x)). ; Assert into the KB
  all(x)(dog(x) => animal(x))

: dog(Fido). ; Assert into the KB
  dog(Fido)

: dog(Fido)?? ; Query assertion without inference
  dog(Fido)
```

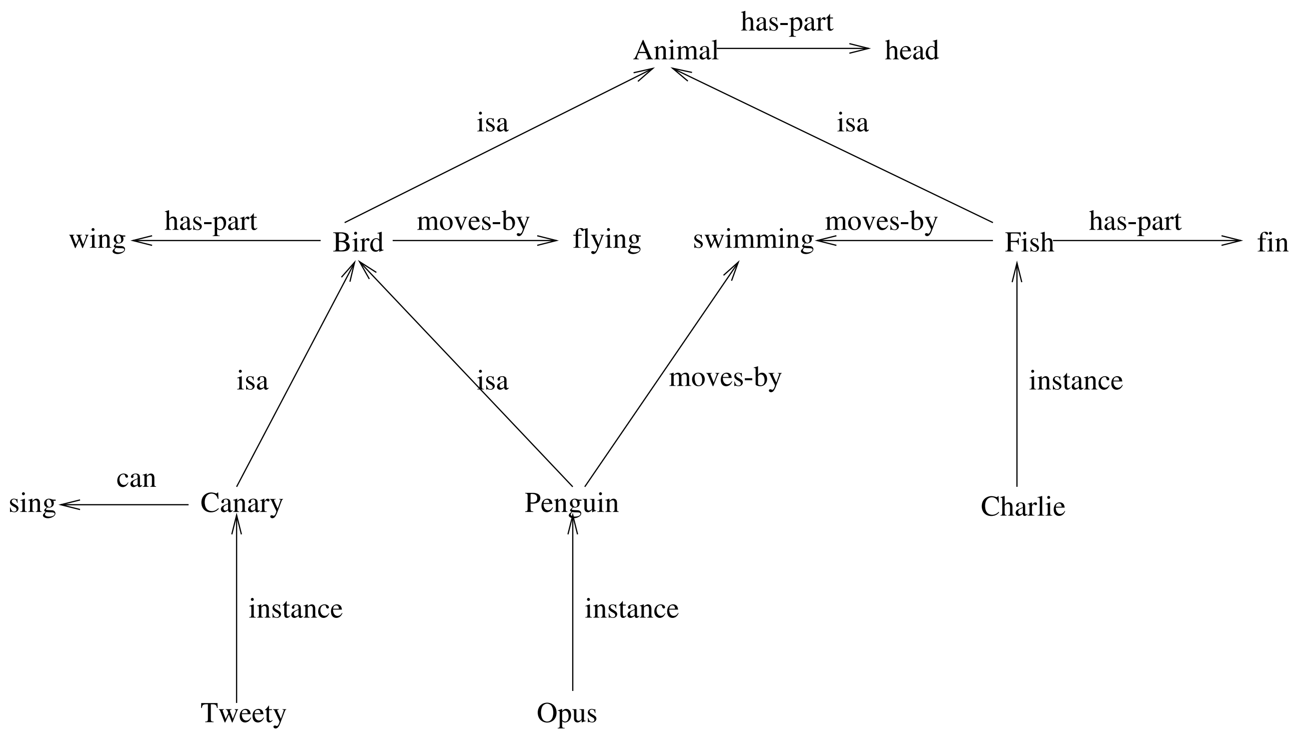
Common SNePSLOG Commands

```
: animal(Fido)?? ; Query assertion without inference

: animal(Fido)? ; Query assertion with inference
  animal(Fido)

: dog(Rover)! ; Assert into the KB & do forward inference
  animal(Rover)
  dog(Rover)

: list-asserted-wffs ; Print all asserted wffs
  animal(Rover)
  dog(Rover)
  animal(Fido)
  dog(Fido)
  all(x)(dog(x) => animal(x))
```

SNePSLOG Demonstrations

```
: demo
Available demonstrations:
  1:  Socrates - Is he mortal?
  2:  UVBR - Demonstrating the Unique Variable Binding Rule
  3:  The Jobs Puzzle - A solution with the Numerical Quantifier
  4:  Pegasus - Why winged horses lead to contradictions
  5:  Schubert's Steamroller
  6:  Rule Introduction - Various examples
  7:  Examples in Chapter 4 of the User's Manual.
  8:  Enter a demo filename
Your choice (q to quit):
```

Semantic Networks



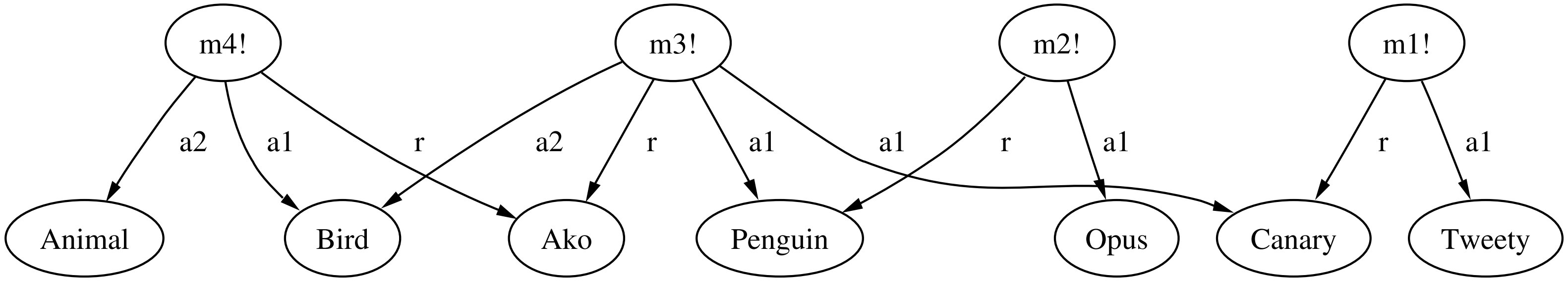
Some psychological evidence.

More efficient search than logical inference.

Unclear semantics.

SNePS as a Network

```
: clearkb
: Canary(Tweety).
: Penguin(Opus).
: Ako({Canary,Penguin}, Bird).
: Ako(Bird, Animal).
: %(show *nodes)
```



Defining Case Frames

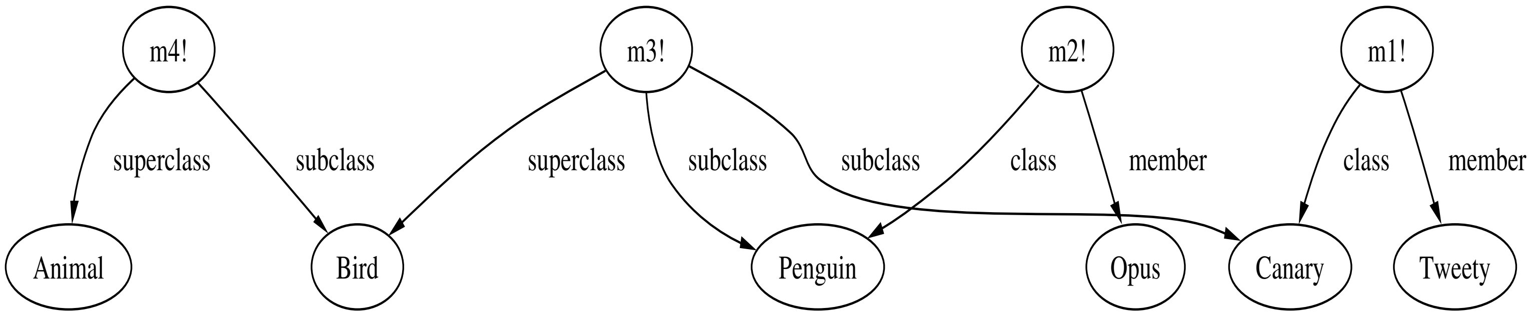
```
: set-mode-3
Net reset
In SNePSLOG Mode 3.
Use define-frame <pred> <list-of-arc-labels>.
...

: define-frame Isa (nil member class)
Isa(x1, x2) will be represented by {<member, x1>, <class, x2>}

: define-frame Ako (nil subclass superclass)
Ako(x1, x2) will be represented by {<subclass, x1>, <superclass, x2>}
```

Entering the KB

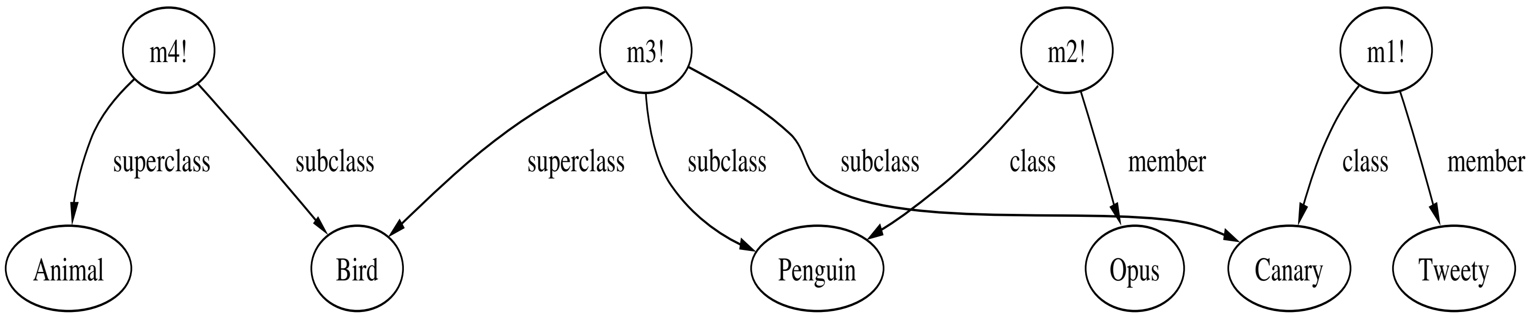
```
: Isa(Tweety, Canary).
  Isa(Tweety,Canary)

: Isa(Opus, Penguin).
  Isa(Opus,Penguin)

: Ako({Canary, Penguin}, Bird)
  Ako({Penguin,Canary},Bird)

: Ako(Bird, Animal).
  Ako(Bird,Animal)
```

The Network

```
: %(show *nodes)
```



Path-Based Inference




```
: %(define-path class
     (compose class (kstar (compose subclass- ! superclass))))
class implied by the path (compose class
                             (kstar
                              (compose subclass- ! superclass)))
class- implied by the path (compose
                              (kstar (compose superclass-
                                              ! subclass))
                              class-)
```

Using Path-Based Inference

```
: list-asserted-wffs
  Ako(Bird,Animal)
  Ako({Penguin,Canary},Bird)
  Isa(Opus,Penguin)
  Isa(Tweety,Canary)

: Isa(Tweety, Animal)?

I wonder if  Isa(Tweety,Animal)
holds within the BS defined by context default-defaultct

I know  Isa(Tweety,Canary)

  Isa(Tweety,Animal)
```