Courteous Logic Programs: Priorized Conflict Handling for Rules

by Benjamin N. Grosof

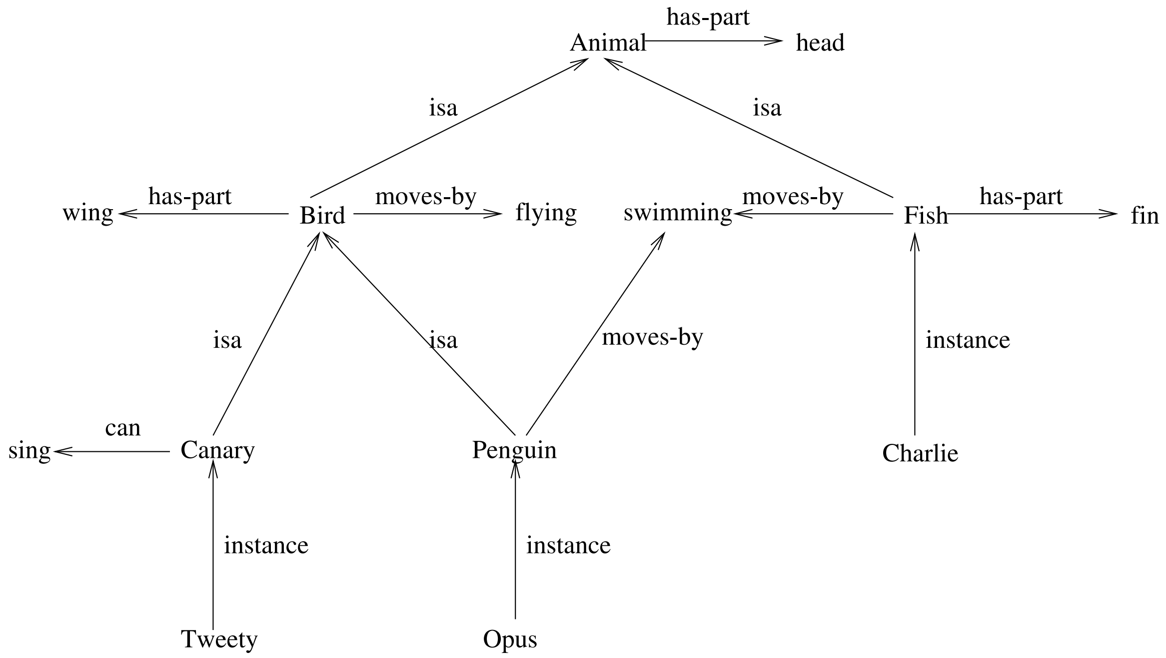
IBM Research Report RC 20836
May 8, 1997; revised Dec. 30, 1997

Presented by
Stuart C. Shapiro
Department of Computer Science and Engineering

April 25, 2002

$\Longrightarrow$ Grosof p. 40

My Motivation



Penguins swimming shadows birds flying by inheritance rules in OO programs.

But what is the declarative (logical) situation?

Default Logic

$$\frac{Bird(x) : Flies(x)}{Flies(x)}$$

$$Penguin(x) \Rightarrow Swims(x)$$

$$Swims(x) \Rightarrow \neg Flies(x)$$

Default rules have a different syntax from "normal" rules, and are even part of the object language.

Basic Idea
Grosof: Courteous Logic Programs

In OO hierarchy, *Penguins swim* **overrides** or **precludes** *Birds fly*.

Why not an $Overrides$ or $Precludes$ meta-predicate that takes "normal" rules?

$\Longrightarrow$ Grosof p. 45

Courteous LP Version of Hierarchy

$$
\begin{array}{ll}
\langle Brd \rangle & Fly(x) \leftarrow Bird(x). \\
 & Bird(x) \leftarrow Canary(x). \\
 & Bird(x) \leftarrow Penguin(x). \\
\langle Peng \rangle & \neg Fly(x) \leftarrow Penguin(x). \\
 & \\
 & Canary(Tweety) \leftarrow . \\
 & Penguin(Opus) \leftarrow . \\
 & \\
 & Overrides(Peng, Brd) \leftarrow . \\
 & \\
\Rightarrow & Fly(Tweety). \\
\Rightarrow & \neg Fly(Opus).
\end{array}
$$

Courteous LP uses Skeptical Reasoning

$\Longrightarrow$ Grosof p. 47

CLP Restrictions

The ground-atom dependency graph of a CLP must be acyclic (no recursion).

$Overrides$ atoms are not allowed in the body of any rule,

only in positive ground facts.

I.e., of the form $Overrides(i, j) \leftarrow .$
where i and j are labels.

This is called a *prioritization fact*.

Prioritization facts must form a strict partial order (transitive, antisymmetric, irreflexive) of the labels.

A prioritization fact may have a label, but it will be ignored.

In the fact $Overrides(i, j) \leftarrow .$, i and j may label rules whose heads have different predicates, but such a fact will be ignored.

Incorrect CLP Version of Hierarchy

$$
\begin{array}{lll}
 & \neg Fly(x) \leftarrow Swim(x). & \\
 & \neg Swim(x) \leftarrow Fly(x). & \textbf{not acyclic} \\
 & & \\
\langle Brd \rangle & Fly(x) \leftarrow Bird(x). & \\
 & Bird(x) \leftarrow Canary(x). & \\
 & Bird(x) \leftarrow Penguin(x). & \\
\langle Peng \rangle & Swim(x) \leftarrow Penguin(x). & \\
 & & \\
 & Canary(Tweety) \leftarrow . & \\
 & Penguin(Opus) \leftarrow . & \\
 & & \\
 & Overrides(Peng, Brd) \leftarrow . & \textbf{different heads}
\end{array}
$$

Operational Example $\Longrightarrow$ Grosof p. 51

Program

$$w \leftarrow \neg u \wedge \sim p.$$
$$\neg u \leftarrow s \wedge \neg t \wedge \sim v.$$
$$q \leftarrow p.$$
$$r \leftarrow \neg p.$$
$$s \leftarrow .$$
$$\neg t \leftarrow .$$
$$p \leftarrow .$$
$$\neg p \leftarrow .$$

As Dependency DAG

$w \leftarrow \neg u \wedge \sim p.$

$\quad\quad \neg u \leftarrow s \wedge \neg t \wedge \sim v.$

$\quad\quad\quad\quad s \leftarrow .$

$\quad\quad\quad\quad \neg t \leftarrow .$

$r \leftarrow \neg p.$

$q \leftarrow p.$

$\quad\quad p \leftarrow .$

$\quad\quad \neg p \leftarrow .$

Program as Dependency DAG

$w \leftarrow \neg u \wedge \sim p.$

$\quad \neg u \leftarrow s \wedge \neg t \wedge \sim v.$

$\qquad s \leftarrow .$

$\qquad \neg t \leftarrow .$

$r \leftarrow \neg p.$

$q \leftarrow p.$

$\quad p \leftarrow .$

$\quad \neg p \leftarrow .$

Sequenced (topologically sorted) & Stratified

$p \leftarrow .$ $\neg p \leftarrow .$
$q \leftarrow p.$
$r \leftarrow \neg p.$
$s \leftarrow .$
$\neg t \leftarrow .$
$\neg u \leftarrow s \wedge \neg t \wedge \sim v.$
$w \leftarrow \neg u \wedge \sim p.$

Finding the Answer Set

	AnswerSet
$p \leftarrow .$ $\neg p \leftarrow .$	
$q \leftarrow p.$	
$r \leftarrow \neg p.$	
$s \leftarrow .$	
$\neg t \leftarrow .$	
$\neg u \leftarrow s \wedge \neg t \wedge \sim v.$	
$w \leftarrow \neg u \wedge \sim p.$	

Finding the Answer Set

	AnswerSet
$p \leftarrow .$ $\neg p \leftarrow .$	$\emptyset$
$q \leftarrow p.$	
$r \leftarrow \neg p.$	
$s \leftarrow .$	
$\neg t \leftarrow .$	
$\neg u \leftarrow s \wedge \neg t \ \wedge \sim v.$	
$w \leftarrow \neg u \ \wedge \sim p.$	

Finding the Answer Set

	AnswerSet
$p \leftarrow .$ $\neg p \leftarrow .$	 $\emptyset$
$q \leftarrow p.$	$\emptyset$
$r \leftarrow \neg p.$	
$s \leftarrow .$	
$\neg t \leftarrow .$	
$\neg u \leftarrow s \wedge \neg t \wedge \sim v.$	
$w \leftarrow \neg u \wedge \sim p.$	

Finding the Answer Set

	AnswerSet
$p \leftarrow .$ $\neg p \leftarrow .$	 $\emptyset$
$q \leftarrow p.$	$\emptyset$
$r \leftarrow \neg p.$	$\emptyset$
$s \leftarrow .$	
$\neg t \leftarrow .$	
$\neg u \leftarrow s \wedge \neg t \wedge \sim v.$	
$w \leftarrow \neg u \wedge \sim p.$	

Finding the Answer Set

	AnswerSet
$p \leftarrow .$ $\neg p \leftarrow .$	 $\emptyset$
$q \leftarrow p.$	$\emptyset$
$r \leftarrow \neg p.$	$\emptyset$
$s \leftarrow .$	s
$\neg t \leftarrow .$	
$\neg u \leftarrow s \wedge \neg t \wedge \sim v.$	
$w \leftarrow \neg u \wedge \sim p.$	

Finding the Answer Set

	AnswerSet
$p \leftarrow .$ $\neg p \leftarrow .$	 $\emptyset$
$q \leftarrow p.$	$\emptyset$
$r \leftarrow \neg p.$	$\emptyset$
$s \leftarrow .$	s
$\neg t \leftarrow .$	$\neg t$
$\neg u \leftarrow s \wedge \neg t \ \wedge \sim v.$	
$w \leftarrow \neg u \ \wedge \sim p.$	

Finding the Answer Set

	AnswerSet
$p \leftarrow .$ $\neg p \leftarrow .$	 $\emptyset$
$q \leftarrow p.$	$\emptyset$
$r \leftarrow \neg p.$	$\emptyset$
$s \leftarrow .$	s
$\neg t \leftarrow .$	$\neg t$
$\neg u \leftarrow s \wedge \neg t \ \wedge \sim v.$	$\neg u$
$w \leftarrow \neg u \ \wedge \sim p.$	

Finding the Answer Set

	AnswerSet
$p \leftarrow .$ $\neg p \leftarrow .$	$\emptyset$
$q \leftarrow p.$	$\emptyset$
$r \leftarrow \neg p.$	$\emptyset$
$s \leftarrow .$	s
$\neg t \leftarrow .$	$\neg t$
$\neg u \leftarrow s \wedge \neg t \wedge \sim v.$	$\neg u$
$w \leftarrow \neg u \wedge \sim p.$	w

Finding the Hierarchy Answer Set

		AnswerSet
	$Overrides(P, B) \leftarrow .$	
	$Canary(T) \leftarrow .$	
	$Penguin(O) \leftarrow .$	
	$Bird(T) \leftarrow Canary(T).$ $Bird(T) \leftarrow Penguin(T).$	
	$Bird(O) \leftarrow Canary(O).$ $Bird(O) \leftarrow Penguin(O).$	
$\langle B \rangle$ $\langle P \rangle$	$Fly(T) \leftarrow Bird(T).$ $\neg Fly(T) \leftarrow Penguin(T).$	
$\langle B \rangle$ $\langle P \rangle$	$Fly(O) \leftarrow Bird(O).$ $\neg Fly(O) \leftarrow Penguin(O).$	

Finding the Hierarchy Answer Set

		AnswerSet
	$Overrides(P, B) \leftarrow .$	$Overrides(P, B)$
	$Canary(T) \leftarrow .$	
	$Penguin(O) \leftarrow .$	
	$Bird(T) \leftarrow Canary(T).$ $Bird(T) \leftarrow Penguin(T).$	
	$Bird(O) \leftarrow Canary(O).$ $Bird(O) \leftarrow Penguin(O).$	
$\langle B \rangle$ $\langle P \rangle$	$Fly(T) \leftarrow Bird(T).$ $\neg Fly(T) \leftarrow Penguin(T).$	
$\langle B \rangle$ $\langle P \rangle$	$Fly(O) \leftarrow Bird(O).$ $\neg Fly(O) \leftarrow Penguin(O).$	

Finding the Hierarchy Answer Set

		AnswerSet
	$Overrides(P,B) \leftarrow .$	$Overrides(P,B)$
	$Canary(T) \leftarrow .$	$Canary(T)$
	$Penguin(O) \leftarrow .$	
	$Bird(T) \leftarrow Canary(T).$ $Bird(T) \leftarrow Penguin(T).$	
	$Bird(O) \leftarrow Canary(O).$ $Bird(O) \leftarrow Penguin(O).$	
$\langle B \rangle$ $\langle P \rangle$	$Fly(T) \leftarrow Bird(T).$ $\neg Fly(T) \leftarrow Penguin(T).$	
$\langle B \rangle$ $\langle P \rangle$	$Fly(O) \leftarrow Bird(O).$ $\neg Fly(O) \leftarrow Penguin(O).$	

Finding the Hierarchy Answer Set

		AnswerSet
	$Overrides(P,B) \leftarrow .$	$Overrides(P,B)$
	$Canary(T) \leftarrow .$	$Canary(T)$
	$Penguin(O) \leftarrow .$	$Penguin(O)$
	$Bird(T) \leftarrow Canary(T).$ $Bird(T) \leftarrow Penguin(T).$	
	$Bird(O) \leftarrow Canary(O).$ $Bird(O) \leftarrow Penguin(O).$	
$\langle B \rangle$ $\langle P \rangle$	$Fly(T) \leftarrow Bird(T).$ $\neg Fly(T) \leftarrow Penguin(T).$	
$\langle B \rangle$ $\langle P \rangle$	$Fly(O) \leftarrow Bird(O).$ $\neg Fly(O) \leftarrow Penguin(O).$	

Finding the Hierarchy Answer Set

		AnswerSet
	$Overrides(P, B) \leftarrow .$	$Overrides(P, B)$
	$Canary(T) \leftarrow .$	$Canary(T)$
	$Penguin(O) \leftarrow .$	$Penguin(O)$
	$Bird(T) \leftarrow Canary(T).$ $Bird(T) \leftarrow Penguin(T).$	$Bird(T)$
	$Bird(O) \leftarrow Canary(O).$ $Bird(O) \leftarrow Penguin(O).$	
$\langle B \rangle$ $\langle P \rangle$	$Fly(T) \leftarrow Bird(T).$ $\neg Fly(T) \leftarrow Penguin(T).$	
$\langle B \rangle$ $\langle P \rangle$	$Fly(O) \leftarrow Bird(O).$ $\neg Fly(O) \leftarrow Penguin(O).$	

Finding the Hierarchy Answer Set

		AnswerSet
	$Overrides(P, B) \leftarrow .$	$Overrides(P, B)$
	$Canary(T) \leftarrow .$	$Canary(T)$
	$Penguin(O) \leftarrow .$	$Penguin(O)$
	$Bird(T) \leftarrow Canary(T).$ $Bird(T) \leftarrow Penguin(T).$	$Bird(T)$
	$Bird(O) \leftarrow Canary(O).$ $Bird(O) \leftarrow Penguin(O).$	$Bird(O)$
$\langle B \rangle$ $\langle P \rangle$	$Fly(T) \leftarrow Bird(T).$ $\neg Fly(T) \leftarrow Penguin(T).$	
$\langle B \rangle$ $\langle P \rangle$	$Fly(O) \leftarrow Bird(O).$ $\neg Fly(O) \leftarrow Penguin(O).$	

Finding the Hierarchy Answer Set

		AnswerSet
	$Overrides(P, B) \leftarrow .$	$Overrides(P, B)$
	$Canary(T) \leftarrow .$	$Canary(T)$
	$Penguin(O) \leftarrow .$	$Penguin(O)$
	$Bird(T) \leftarrow Canary(T).$ $Bird(T) \leftarrow Penguin(T).$	$Bird(T)$
	$Bird(O) \leftarrow Canary(O).$ $Bird(O) \leftarrow Penguin(O).$	$Bird(O)$
$\langle B \rangle$ $\langle P \rangle$	$Fly(T) \leftarrow Bird(T).$ $\neg Fly(T) \leftarrow Penguin(T).$	$Fly(T)$
$\langle B \rangle$ $\langle P \rangle$	$Fly(O) \leftarrow Bird(O).$ $\neg Fly(O) \leftarrow Penguin(O).$	

Finding the Hierarchy Answer Set

		AnswerSet
	$Overrides(P, B) \leftarrow .$	$Overrides(P, B)$
	$Canary(T) \leftarrow .$	$Canary(T)$
	$Penguin(O) \leftarrow .$	$Penguin(O)$
	$Bird(T) \leftarrow Canary(T).$ $Bird(T) \leftarrow Penguin(T).$	$Bird(T)$
	$Bird(O) \leftarrow Canary(O).$ $Bird(O) \leftarrow Penguin(O).$	$Bird(O)$
$\langle B \rangle$ $\langle P \rangle$	$Fly(T) \leftarrow Bird(T).$ $\neg Fly(T) \leftarrow Penguin(T).$	$Fly(T)$
$\langle B \rangle$ $\langle P \rangle$	$Fly(O) \leftarrow Bird(O).$ $\neg Fly(O) \leftarrow Penguin(O).$	$\neg Fly(O)$

Formal Constructive Definition of Answer Set

Let ρ be total (ground) atom stratification of CLP, s.t. prioritization atoms come first.

Let p_i be i^{th} atom in ρ.

Answer set S:

$$S_0 = \emptyset$$
$$S_i = \bigcup_{j=1,\ldots,i} T_j \quad i \geq 1$$
$$S = \bigcup_i T_i$$

$$\begin{aligned}
T_i = \{\sigma p_i \;\mid\; & Cand_i^{\sigma} \neq \emptyset \;, \\
& \forall k \in Cand_i^{\neg\sigma} \\
& .\ \exists j \in Cand_i^{\sigma} \\
& .\ S_{i-1} \models Overrides(j,k)\} \\
Cand_i^{\sigma} = \{j \;\mid\; & labels(j,r) \;, \\
& Head(r) = \sigma p_i \;, \\
& S_{i-1} \models Body(r)\}
\end{aligned}$$

σ is classical sign, either $+$ or - ($\neg$).

[Grosof, p 6–7]

Some Results

$\Longrightarrow$ Grosof p. 52

CLP Gets Effect of CWA

$$
\begin{array}{ll}
 & Overrides(sched, cwa) \\
\langle cwa \rangle & \neg flight(S, D, T, A) \\
\langle sched \rangle & flight(miami, detroit, 10am, elysian\_air) \leftarrow \\
\langle sched \rangle & flight(jfk, new\_orleans, 4am, fountain\_air) \leftarrow \\
\langle sched \rangle & flight(dallas, seattle, 7am, middle\_air) \leftarrow
\end{array}
$$

[Grosof, p. 22]

Preservation in Parallel Merging

For example & problem $\Longrightarrow$ Grosof p. 54

CLP solution:

		AnswerSet
	$Overrides(1,2) \leftarrow .$ $Overrides(4,3) \leftarrow .$	
$\langle 1 \rangle$	$a \leftarrow .$	
$\langle 2 \rangle$	$b \leftarrow .$	
$\langle 3 \rangle$	$c \leftarrow .$	
$\langle 4 \rangle$	$d \leftarrow .$	
$\langle 1 \rangle$ $\langle 4 \rangle$ $\langle 2 \rangle$ $\langle 3 \rangle$	$p \leftarrow a.$ $p \leftarrow d.$ $\neg p \leftarrow b.$ $\neg p \leftarrow c.$	

Preservation in Parallel Merging

For example & problem $\Longrightarrow$ Grosof p. 54

CLP solution:

		AnswerSet
	$Overrides(1,2) \leftarrow .$ $Overrides(4,3) \leftarrow .$	$Overrides(1,2)$ $Overrides(4,3)$
$\langle 1 \rangle$	$a \leftarrow .$	
$\langle 2 \rangle$	$b \leftarrow .$	
$\langle 3 \rangle$	$c \leftarrow .$	
$\langle 4 \rangle$	$d \leftarrow .$	
$\langle 1 \rangle$ $\langle 4 \rangle$ $\langle 2 \rangle$ $\langle 3 \rangle$	$p \leftarrow a.$ $p \leftarrow d.$ $\neg p \leftarrow b.$ $\neg p \leftarrow c.$	

Preservation in Parallel Merging

For example & problem $\Longrightarrow$ Grosof p. 54

CLP solution:

		AnswerSet
	$Overrides(1,2) \leftarrow .$ $Overrides(4,3) \leftarrow .$	$Overrides(1,2)$ $Overrides(4,3)$
$\langle 1 \rangle$	$a \leftarrow .$	a
$\langle 2 \rangle$	$b \leftarrow .$	b
$\langle 3 \rangle$	$c \leftarrow .$	c
$\langle 4 \rangle$	$d \leftarrow .$	d
$\langle 1 \rangle$ $\langle 4 \rangle$ $\langle 2 \rangle$ $\langle 3 \rangle$	$p \leftarrow a.$ $p \leftarrow d.$ $\neg p \leftarrow b.$ $\neg p \leftarrow c.$	

Preservation in Parallel Merging

For example & problem $\Longrightarrow$ Grosof p. 54

CLP solution:

		AnswerSet
	$Overrides(1,2) \leftarrow .$ $Overrides(4,3) \leftarrow .$	$Overrides(1,2)$ $Overrides(4,3)$
$\langle 1 \rangle$	$a \leftarrow .$	a
$\langle 2 \rangle$	$b \leftarrow .$	b
$\langle 3 \rangle$	$c \leftarrow .$	c
$\langle 4 \rangle$	$d \leftarrow .$	d
$\langle 1 \rangle$ $\langle 4 \rangle$ $\langle 2 \rangle$ $\langle 3 \rangle$	$p \leftarrow a.$ $p \leftarrow d.$ $\neg p \leftarrow b.$ $\neg p \leftarrow c.$	p

Adding Prioritized Conflict Handling
to SNePS 2.5

A M.S. thesis project by Bharat Bhushan
under the supervision of Stuart C. Shapiro.

SNePS 2.5 is a product of
the SNePS Research Group.

SNePS 2.5 Basics

A *hypothesis* is a proposition asserted to SNePS by the user.

A *context* is a set of hypotheses.

There is always a *current context, CC.*

Every proposition, p has a set of *origin sets,* $os(p)$.

An origin set is a set of hypotheses.

A proposition, p is *asserted* in a context, c iff

$$\exists s \in os(p)[\forall h \in s[h \in c]]$$

Modifications for Prioritized Conflict Handling

A proposition, p, is *derivable* in a context, c ($Derivable(p, c)$), iff

$$\exists s \in os(p)[\forall h \in s[h \in c]]$$

There is a special predicate $Precludes(p1, p2)$ that may hold between two propositions.
$Precludes$ predicates may be included in rules, like other predicates.

A proposition, p is *asserted* in a context, c ($Asserted(p, c)$), iff

$$\begin{aligned}
&Derivable(p, c) \\
&\quad \wedge[\neg Derivable(\neg p, c) \\
&\qquad \vee Asserted(Precludes(p, \neg p), c) \\
&\qquad \vee[\exists s \in os(p) \\
&\qquad\quad \neg\exists s' \in os(\neg p) \\
&\qquad\qquad (\exists h \in s(\exists h' \in s' \\
&\qquad\qquad\qquad Asserted(Precludes(h', h), c)))]]
\end{aligned}$$

A proposition, p, is *asserted*
iff it is asserted in CC.