Answer Set Semantics vs. Information Term Semantics

Camillo Fiorentini, Mario Ornaghi

Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, Italy

September 13, 2007

The context of this talk

Work in progress: a DLV* implementation of CooML (Constructive object oriented Modeling Language) snapshot generation.

In this talk: ideas/results from our work in progress comparing CooML snapshot semantics and answer sets semanitcs.

The Context: Representing OO Modeling Languages in DLV

CooML (Constructive object oriented Modeling Language):

- The novelty: semantics based on Fcl
  (an intermediate constructive logic, Miglioli [1989], similar Medvedev's Logic of finite problems [1962]).
- *Logic based tools*:
  - ...
  - **snapshot generation for model validation**

The Context: Snapshot Generation for Model Validation

To validate an OO model M w.r.t. its informal requirements,
show human viable snapshots (representing system states)
Remark: must be empirical

- UML+OCL Example: USE (Gogolla 2004), imperative generation language
- In CooML: snapshots
  - specified by *class properties*
  - through *pieces of information*
  - declarative generation
  - UML+OCL can be represented in CooML

Outline

The syntax of class properties

Class Property (F a first order formula, G a class generator, T the classical truth operator):

Simple S	::=	literal $\mid$ $T(F)$
Existential E	::=	$S \mid E \wedge E \mid E \vee E \mid \exists \underline{x}\, E$
Class property P	::=	$E \mid \forall \underline{x}.G \rightarrow E$

The syntax of class properties

Class Property (F a first order formula, G a class generator, T the classical truth operator):

Simple S	$::=$	literal $\mid$ $T(F)$
Existential E	$::=$	$S \mid E \wedge E \mid E \vee E \mid \exists \underline{x}\, E$
Class property P	$::=$	$E \mid \forall \underline{x}.G \rightarrow E$

Example

A receipt is empty (no items), or it is not empty and computes a grand total. Each item has a price. The grand total is the sum of the prices of all the items.

class : $\forall x.receipt(x) \rightarrow T(\neg \exists i.item(i,x))$

The syntax of class properties

Class Property (F a first order formula, G a class generator, T the classical truth operator):

Simple S	::=	literal $\mid$ $T(F)$
Existential E	::=	S $\mid$ $E \wedge E$ $\mid$ $E \vee E$ $\mid$ $\exists \underline{x}\, E$
Class property P	::=	E $\mid$ $\forall \underline{x}.G \rightarrow E$

Example

A receipt is empty (no items), or it is not empty and computes a grand total. Each item has a price. The grand total is the sum of the prices of all the items.

class : $\forall x.receipt(x) \rightarrow T(\neg \exists i.item(i,x))$
$\vee ((\exists t.t = total(x)) \wedge T(\exists i.item(i,x)))$

The syntax of class properties

Class Property (F a first order formula, G a class generator, T the classical truth operator):

Simple S	::=	literal $\mid$ $T(F)$
Existential E	::=	S $\mid$ $E \wedge E$ $\mid$ $E \vee E$ $\mid$ $\exists \underline{x}\ E$
Class property P	::=	E $\mid$ $\forall \underline{x}.G \rightarrow E$

Example

A receipt is empty (no items), or it is not empty and computes a grand total. Each item has a price. The grand total is the sum of the prices of all the items.

class : $\forall x.receipt(x) \rightarrow T(\neg \exists i.item(i,x)) \vee ((\exists t.t = total(x)) \wedge T(\exists i.item(i,x)))$

class : $\forall i,x.item(i,x) \rightarrow receipt(x) \wedge \exists p.p = price(i)$

The syntax of class properties

Class Property (F a first order formula, G a class generator, T the classical truth operator):

Simple S	::=	literal $\mid$ $T(F)$
Existential E	::=	$S \mid E \wedge E \mid E \vee E \mid \exists \underline{x}\, E$
Class property P	::=	$E \mid \forall \underline{x}.G \rightarrow E$

Example

A receipt is empty (no items), or it is not empty and computes a grand total. Each item has a price. The grand total is the sum of the prices of all the items.

class : $\forall x.receipt(x) \rightarrow T(\neg \exists i.item(i,x)) \vee ((\exists t.t = total(x)) \wedge T(\exists i.item(i,x)))$

class : $\forall i,x.item(i,x) \rightarrow receipt(x) \wedge \exists p.p = price(i)$

constr : $T(\forall x.receipt(x) \rightarrow total(x) = sum(price(i) : item(i,x)))$

Snapshot Semantics: an example of piece of information

$$[\,[\, r_1, [\,[2, [\,[12, \top], \top]\,]\,], [\, r_2, [1, \top]\,]\,]\,] \;:$$

$$\forall x.receipt(x) \rightarrow T(\neg \exists i.item(i, x)) \\ \vee((\exists t.t = total(x)) \wedge T(\exists i.item(i, x)))$$

Snapshot Semantics: an example of piece of information

$$[\,[\,r_1, [\,[2, [\,[12, \top], \top]\,]\,], [\,r_2, [1, \top]\,]\,] \;:$$
$$\forall x.receipt(x) \rightarrow T(\neg \exists i.item(i, x))$$
$$\vee((\exists t.t = total(x)) \wedge T(\exists i.item(i, x)))$$

Receipt: r_1

Snapshot Semantics: an example of piece of information

$$[\,[\,r_1, [\,[2, [\,[12, \mathrm{T}], \mathrm{T}]\,]\,], [\,r_2, [1, \mathrm{T}]\,]\,]\ :$$

$$\forall x.receipt(x) \rightarrow T(\neg \exists i.item(i, x)) \lor ((\exists t.t = total(x)) \land T(\exists i.item(i, x)))$$

Receipt: r_1

Snapshot Semantics: an example of piece of information

$$[\,[\, r_1, [\,[2, [\,[12, \mathrm{T}], \mathrm{T}]\,]\,], [\, r_2, [1, \mathrm{T}]\,]\,] \; :$$

$$\forall x.receipt(x) \rightarrow T(\neg \exists i.item(i,x)) \\ \qquad \vee((\exists t.t = total(x)) \wedge T(\exists i.item(i,x)))$$

Receipt: r_1

Total: 12

Snapshot Semantics: an example of piece of information

$$[\,[\,r_1, [\,[2, [\,[12, \mathrm{T}], \mathrm{T}]\,]\,], [\,r_2, [1, \mathrm{T}]\,]\,] :$$
$$\forall x.receipt(x) \rightarrow T(\neg \exists i.item(i,x))$$
$$\vee((\exists t.t = total(x)) \wedge T(\exists i.item(i,x)))$$

Receipt: r_1

Item	Price
it_1:	5
it_2:	7

Total: 12

Snapshot Semantics: an example of piece of information

$$[\,[\, r_1, [\,[2, [\,[12, \top], \top]\,]\,], [\, r_2, [1, \top]\,]\,] \;:$$

$$\forall x.receipt(x) \rightarrow T(\neg \exists i.item(i, x)) \vee ((\exists t.t = total(x)) \wedge T(\exists i.item(i, x)))$$

Receipt: r_1

Item	Price
it_1:	5
it_2:	7

Total: 12

Receipt: r_2

Snapshot Semantics: an example of piece of information

$[\,[\,r_1, [\,[2, [\,[12, \mathrm{T}], \mathrm{T}]\,]\,], [\,r_2, [1, \mathrm{T}]\,]\,]$:

$$\forall x.receipt(x) \rightarrow T(\neg \exists i.item(i,x)) \vee ((\exists t.t = total(x)) \wedge T(\exists i.item(i,x)))$$

Receipt: r_1

Item	Price
it_1:	5
it_2:	7

Total: 12

Receipt: r_2

Snapshot Semantics: pieces of information and their "answer sets"

Piece of information $\tau : P$	its I-answer set $ans(\tau : P)$
$\mathrm{T} : S$	$\{S\}$
$[\tau_1, \tau_2] : P_1 \wedge P_2$ with $\tau_1 : P_1$ and $\tau_2 : P_2$	$ans(\tau_1 : P_1) \cup ans(\tau_2 : P_2)$
$[k, \tau] : P_1 \vee P_2$ with $k \in 1..2, \tau : P_k$	$ans(\tau : P_k)$
$[\underline{v}, \tau] : \exists \underline{x}\, P$ with $\underline{v}$ values for $\underline{x}$, $\tau : P$	$ans(\tau : P(\underline{v}))$
$[\,[\underline{v}_1, \tau_1], \ldots, [\underline{v}_n, \tau_n]\,] : \forall \underline{x}.G \rightarrow P$ $n \geq 0$, $\underline{v}_j$ val. $\underline{x}$, $\tau_j \in \mathrm{IT}(P)$	$\bigcup_j ans(\tau_j : P(\underline{v}_j)) \cup$ $\{\, \forall (G(\underline{x}) \leftrightarrow \underline{x} \in [\underline{v}_1, \ldots, \underline{v}_n])\,\}$

Snapshot Semantics: pieces of information and their "answer sets"

Piece of information $\tau : P$	its I-answer set $ans(\tau : P)$
$\mathrm{T} : S$	$\{S\}$
$[\tau_1, \tau_2] : P_1 \wedge P_2$ with $\tau_1 : P_1$ and $\tau_2 : P_2$	$ans(\tau_1 : P_1) \cup ans(\tau_2 : P_2)$
$[k, \tau] : P_1 \vee P_2$ with $k \in 1..2, \tau : P_k$	$ans(\tau : P_k)$
$[\underline{v}, \tau] : \exists \underline{x}\, P$ with $\underline{v}$ values for $\underline{x}$, $\tau : P$	$ans(\tau : P(\underline{v}))$
$[\,[\underline{v}_1, \tau_1], \ldots, [\underline{v}_n, \tau_n]\,] : \forall \underline{x}.G \rightarrow P$ $n \geq 0$, $\underline{v}_j$ val. $\underline{x}$, $\tau_j \in \mathrm{IT}(P)$	$\bigcup_j ans(\tau_j : P(\underline{v}_j)) \cup \{\, \forall(G(\underline{x}) \leftrightarrow \underline{x} \in [\underline{v}_1, \ldots, \underline{v}_n])\,\}$

Snapshot Semantics: pieces of information and their "answer sets"

Piece of information $\tau : P$	its I-answer set $ans(\tau : P)$
$\mathrm{T} : S$	$\{S\}$
$[\tau_1, \tau_2] : P_1 \wedge P_2$ with $\tau_1 : P_1$ and $\tau_2 : P_2$	$ans(\tau_1 : P_1) \cup ans(\tau_2 : P_2)$
$[k, \tau] : P_1 \vee P_2$ with $k \in 1..2, \tau : P_k$	$ans(\tau : P_k)$
$[\underline{v}, \tau] : \exists \underline{x}\, P$ with $\underline{v}$ values for $\underline{x}$, $\tau : P$	$ans(\tau : P(\underline{v}))$
$[\,[\underline{v}_1, \tau_1], \ldots, [\underline{v}_n, \tau_n]\,] : \forall \underline{x}.G \rightarrow P$ $n \geq 0$, $\underline{v}_j$ val. $\underline{x}$, $\tau_j \in \mathrm{IT}(P)$	$\bigcup_j ans(\tau_j : P(\underline{v}_j)) \cup$ $\{\, \forall (G(\underline{x}) \leftrightarrow \underline{x} \in [\underline{v}_1, \ldots, \underline{v}_n])\,\}$

Snapshot Semantics: pieces of information and their "answer sets"

Piece of information $\tau : P$	its I-answer set $ans(\tau : P)$
$\mathrm{T} : S$	$\{S\}$
$[\tau_1, \tau_2] : P_1 \wedge P_2$ with $\tau_1 : P_1$ and $\tau_2 : P_2$	$ans(\tau_1 : P_1) \cup ans(\tau_2 : P_2)$
$[k, \tau] : P_1 \vee P_2$ with $k \in 1..2, \tau : P_k$	$ans(\tau : P_k)$
$[\underline{v}, \tau] : \exists \underline{x}\ P$ with $\underline{v}$ values for $\underline{x}$, $\tau : P$	$ans(\tau : P(\underline{v}))$
$[\,[\underline{v}_1, \tau_1], \ldots, [\underline{v}_n, \tau_n]\,] : \forall \underline{x}.G \rightarrow P$ $n \geq 0$, $\underline{v}_j$ val. $\underline{x}$, $\tau_j \in \mathrm{IT}(P)$	$\bigcup_j ans(\tau_j : P(\underline{v}_j)) \ \cup$ $\{\ \forall(G(\underline{x}) \ \leftrightarrow\ \underline{x} \in [\underline{v}_1, \ldots, \underline{v}_n])\ \}$

Snapshot Semantics: pieces of information and their "answer sets"

Piece of information $\tau : P$	its I-answer set $ans(\tau : P)$
$\mathrm{T} : S$	$\{S\}$
$[\tau_1, \tau_2] : P_1 \wedge P_2$ with $\tau_1 : P_1$ and $\tau_2 : P_2$	$ans(\tau_1 : P_1) \cup ans(\tau_2 : P_2)$
$[k, \tau] : P_1 \vee P_2$ with $k \in 1..2, \tau : P_k$	$ans(\tau : P_k)$
$[\underline{v}, \tau] : \exists \underline{x}\, P$ with $\underline{v}$ values for $\underline{x}$, $\tau : P$	$ans(\tau : P(\underline{v}))$
$[\,[\underline{v}_1, \tau_1], \ldots, [\underline{v}_n, \tau_n]\,] : \forall \underline{x}.G \rightarrow P$ $n \geq 0$, $\underline{v}_j$ val. $\underline{x}$, $\tau_j \in \mathrm{IT}(P)$	$\bigcup_j ans(\tau_j : P(\underline{v}_j)) \cup$ $\{\, \forall (G(\underline{x}) \leftrightarrow \underline{x} \in [\underline{v}_1, \ldots, \underline{v}_n])\,\}$

Example: pieces of information and their I-answer sets

Pieces of information:

$$[\,[\, r_1, [\,[2, [\,[12, \mathrm{T}], \mathrm{T}]\,]\,],\ [\, r_2, [1, \mathrm{T}]\,]\,]\ :$$
$$\forall x. receipt(x) \rightarrow T(\neg\exists i. item(i,x)) \lor ((\exists t. t = total(x)) \land T(\exists i. item(i,x)))$$

I-answer sets:

Example: pieces of information and their I-answer sets

Pieces of information:

$$[\,[\,r_1, [\,[2, [\,[12, \top], \top]\,]\,], \,[\,r_2, [1, \top]\,]\,] \;:$$
$$\forall x.receipt(x) \rightarrow T(\neg\exists i.item(i,x)) \vee ((\exists t.t = total(x)) \wedge T(\exists i.item(i,x)))$$

I-answer sets:

$$receipt(x) \;\leftrightarrow\; x \in [\,r_1,\; r_2\,] \quad \text{(class generator axiom)}$$

Example: pieces of information and their I-answer sets

Pieces of information:

$$[\,[\,r_1, [\,[2, [\,[12, \mathrm{T}], \mathrm{T}]\,]\,], \;[\,r_2, [1, \mathrm{T}]\,]\,] \;:$$
$$\forall x.receipt(x) \rightarrow T(\neg\exists i.item(i,x)) \lor ((\exists t.t = total(x)) \land T(\exists i.item(i,x)))$$

I-answer sets:

$$receipt(x) \;\leftrightarrow\; x \in [\,r_1,\; r_2\,] \quad \text{(class generator axiom)}$$

Example: pieces of information and their I-answer sets

Pieces of information:

$$[\,[\,r_1, [\,[2, [\,[12, \mathrm{T}], \mathrm{T}]\,]\,], \;[\,r_2, [1, \mathrm{T}]\,]\,]\;:$$
$$\forall x.receipt(x) \rightarrow T(\neg\exists i.item(i,x))$$
$$\vee((\exists t.t = total(x)) \wedge T(\exists i.item(i,x)))$$

I-answer sets:

$$receipt(x) \;\leftrightarrow\; x \in [\,r_1,\, r_2\,] \quad \text{(class generator axiom)}$$
$$12 = total(r_1),$$

Example: pieces of information and their I-answer sets

Pieces of information:

$$[\,[\,r_1, [\,[2, [\,[12, \mathrm{T}], \mathrm{T}]\,]\,], \ [\,r_2, [1, \mathrm{T}]\,]\,]\ :$$
$$\forall x.receipt(x) \rightarrow T(\neg\exists i.item(i,x))$$
$$\vee((\exists t.t = total(x)) \wedge T(\exists i.item(i,x)))$$

I-answer sets:

$$receipt(x) \ \leftrightarrow \ x \in [\,r_1, \ r_2\,] \quad \text{(class generator axiom)}$$
$$12 = total(r_1), T(\exists i.item(i, r_1)),$$

Example: pieces of information and their I-answer sets

Pieces of information:

$$[\,[\,r_1, [\,[2, [\,[12, \mathrm{T}], \mathrm{T}]\,]\,], \,[\,r_2, [1, \mathrm{T}]\,]\,] \;:$$
$$\forall x.receipt(x) \rightarrow T(\neg \exists i.item(i,x)) \\ \lor((\exists t.t = total(x)) \land T(\exists i.item(i,x)))$$

I-answer sets:

$$receipt(x) \;\leftrightarrow\; x \in [\,r_1,\, r_2\,] \quad \text{(class generator axiom)}$$
$$12 = total(r_1),\, T(\exists i.item(i, r_1)), T(\neg \exists i.item(i, r_2))$$

Example: pieces of information and their I-answer sets

Pieces of information:

$$[\,[\,r_1, [\,[2, [\,[12, \mathrm{T}], \mathrm{T}]\,]\,],\; [\,r_2, [1, \mathrm{T}]\,]\,] \;:$$
$$\forall x.receipt(x) \rightarrow T(\neg\exists i.item(i,x)) \\ \vee((\exists t.t = total(x)) \wedge T(\exists i.item(i,x)))$$

$$[\,[\,(it_1, r_1), [5, \mathrm{T}]\,],\; [\,(it_2, r_1), [7, \mathrm{T}]\,]\,] :$$
$$\forall i, x.item(i,x) \rightarrow \exists p.p = price(i)$$

I-answer sets:

$$receipt(x) \;\leftrightarrow\; x \in [\,r_1,\; r_2\,] \quad \text{(class generator axiom)}$$
$$12 = total(r_1), T(\exists i.item(i, r_1)), T(\neg\exists i.item(i, r_2))$$
$$item(i,x) \;\leftrightarrow\; (i,x) \in [\,(it_1, r_1), (it_2, r_1)\,]$$
$$5 = price(it_1),\; 7 = price(it_2)$$

Example: pieces of information and their I-answer sets

Pieces of information:

$$[\,[\,r_1, [\,[2, [\,[12, \top], \top]\,]\,],\ [\,r_2, [1, \top]\,]\,]\ :$$
$$\forall x.receipt(x) \rightarrow T(\neg\exists i.item(i,x)) \vee ((\exists t.t = total(x)) \wedge T(\exists i.item(i,x)))$$

$$[\,[\,(it_1, r_1), [5, \top]\,],\ [\,(it_2, r_1), [7, \top]\,]\,]:$$
$$\forall i, x.item(i,x) \rightarrow \exists p.p = price(i)$$

$$t : T(\,\forall x.receipt(x) \ \rightarrow\ total(x) = sum(price(i) : item(i,x))\,)$$

I-answer sets:

$$receipt(x) \ \leftrightarrow\ x \in [\,r_1,\ r_2\,] \quad \text{(class generator axiom)}$$
$$12 = total(r_1),\ T(\exists i.item(i,r_1)), T(\neg\exists i.item(i,r_2))$$
$$item(i,x) \ \leftrightarrow\ (i,x) \in [\,(it_1, r_1),\ (it_2, r_1)\,]$$
$$5 = price(it_1),\ 7 = price(it_2)$$
$$T(\,\forall x.receipt(x) \ \rightarrow\ total(x) = sum(price(i) : item(i,x))\,)$$

CooML Consistency

CooML consistency: intended models(atoms + generator axioms) $\models$ T-Formulas)

$\Rightarrow$ consistency

Example

$$5 = price(it_1),\ 7 = price(it_2),\ 12 = total(r_1)$$
$$\forall(receipt(x)\ \leftrightarrow\ x \in [\, r_1,\ r_2\,]),$$
$$\forall(item(i,x)\ \leftrightarrow\ (i,x) \in [\,(it_1, r_1),\ (it_2, r_1)\,])$$

$$\models$$

$$T\ (\forall x.receipt(x)\ \rightarrow\ total(x) = sum(price(i) : item(i,x))\ )$$
$$T(\exists i.item(i, r_1)),\ \ T(\neg\exists i.item(i, r_2))$$

Representing CooML models as DLV programs

Starting from

Example

$$\text{class}: \quad T(\neg\exists i.item(i,X)) \lor ((\exists t.total(X,t)) \land T(\exists i.item(i,X))) \leftarrow receipt(X).$$

$$\text{class}: \quad receipt(X) \land \exists p.p = price(I) \leftarrow item(I,X).$$

...

replace T-formulas by atoms + constraints

Representing CooML models as DLV programs

replacing $T(\neg\exists I.item(I,X))$ by $empty(X)$:

Example

class : $empty(X) \vee ((\exists t.total(X,t)) \wedge -empty(X))$
$\leftarrow receipt(X).$

class : $receipt(X) \wedge \exists p.p = price(I)$
$\leftarrow item(I,X).$

constr : $false \leftarrow \exists I.empty(X) \wedge item(I,X)$

...

... further translation ...: DLV programs such that CooML snapshots are represented by stable models.

Representing CooML models as nested programs with ∃

But what happens if we start directly from:

Example

$$\text{class}: \quad \neg(\exists i.item(i,X)) \ \lor ((\exists t.total(X,t)) \land \neg\neg(\exists i.item(i,X))) \leftarrow receipt(X).$$

$$\text{class}: \quad receipt(X) \ \land \ \exists p.p = price(I) \leftarrow item(I,X).$$

...

Stable Model Semantics vs Pieces of Information (*Fcl*) Semantics.

We started from the propositional case

Outline

The *Fcl* propositional logic

Propositional syntax, where simple formulas are atomic or negated *(can be extended with Nelson's negation)*

Piece of information $\tau : P$	I-answer set $ans(\tau : P)$
$\mathrm{T} : S$	$\{S\}$
$[\tau_1, \tau_2] : P_1 \wedge P_2$	$ans(\tau_1 : P_1) \cup ans(\tau_2 : P_2)$
$[k, \tau] : P_1 \vee P_2$	$ans(\tau : P_k)$

The *Fcl* propositional logic

Propositional syntax, where simple formulas are atomic or negated *(can be extended with Nelson's negation)*

Piece of information $\tau : P$	I-answer set $ans(\tau : P)$
$\mathrm{T} : S$	$\{S\}$
$[\tau_1, \tau_2] : P_1 \wedge P_2$	$ans(\tau_1 : P_1) \cup ans(\tau_2 : P_2)$
$[k, \tau] : P_1 \vee P_2$	$ans(\tau : P_k)$
$f : P_1 \rightarrow P_2$ with $f : \mathrm{IT}(P_1) \rightarrow \mathrm{IT}(P_2)$	$\bigcup_{\tau \in \mathrm{IT}(P_1)} (ans(\tau : P_1) \rightarrow ans(f(\tau) : P_2))$

On the Semantics of *Fcl*

Theorem (Relationship with classical logic)

A classical interpretation $I \models P$ *iff there is* $\tau : P$ *such that* $I \models ans(\tau : P)$.

Definition (Constructive validity)

P is *Fcl*-valid iff there is $\tau : P$ such that for every interpretation I, $I \models ans(\tau : P)$.

On the Semantics of *Fcl*

Theorem (Relationship with classical logic)

A classical interpretation $I \models P$ *iff there is* $\tau : P$ *such that* $I \models ans(\tau : P)$.

Definition (Constructive validity)

P is *Fcl*-valid iff there is $\tau : P$ such that for every interpretation I, $I \models ans(\tau : P)$.

Example

$$(\neg\neg p \vee \neg p) \;\wedge\; (\neg\neg p \rightarrow a) \;\wedge\; (\neg p \rightarrow b) \;\;\rightarrow\;\; a \vee b$$

map τ: $[\,[1, \mathrm{T}], \lambda x.\mathrm{T}, \lambda x.\mathrm{T}\,] \;\mapsto_\tau\; [1, \mathrm{T}]$,
$[\,[2, \mathrm{T}], \lambda x.\mathrm{T}, \lambda x.\mathrm{T}\,] \;\mapsto_\tau\; [2, \mathrm{T}]$

$$(\neg\neg p \rightarrow a) \;\wedge\; (\neg p \rightarrow b) \;\;\rightarrow\;\; a \vee b$$

no valid τ

On the Semantics of *Fcl*

Theorem (Relationship with classical logic)

A classical interpretation $I \models P$ *iff there is* $\tau : P$ *such that* $I \models ans(\tau : P)$.

Definition (Constructive validity)

P is *Fcl*-valid iff there is $\tau : P$ such that for every interpretation I, $I \models ans(\tau : P)$.

Example

On the other hand

$$(\neg\neg p \rightarrow a) \ \wedge \ (\neg p \rightarrow b) \quad \rightarrow \quad a \vee b$$

holds in "Here and There".

Are there pieces of information for stable models?

Fcl	FHT
Interpretation $I \mapsto \tau_I : F$	Stable model $M \mapsto \tau_M : F$
Truth preserving maps	maps ...

Answer sets for nested expressions using pieces of information

Basic concepts:

- Positive answers: $ans^{+}(\tau : F) = \{p \mid p \in ans(\tau : F)\}$,
- Negative answers: $ans^{-}(\tau : F) = \{\neg H \mid \neg H \in ans(\tau : F)\}$

Answer sets for nested expressions using pieces of information

Basic concepts:

- Positive answers: $ans^{+}(\tau : F) = \{p \mid p \in ans(\tau : F)\}$,
- Negative answers: $ans^{-}(\tau : F) = \{\neg H \mid \neg H \in ans(\tau : F)\}$
- Default interpretation $\mathrm{DI}(\tau : F)$: $\mathrm{DI}(\tau : F) \models p$ iff $p \in ans^{+}(\tau : F)$

Answer sets for nested expressions using pieces of information

Basic concepts:

- Positive answers: $ans^+(\tau : F) = \{p \mid p \in ans(\tau : F)\}$,
- Negative answers: $ans^-(\tau : F) = \{\neg H \mid \neg H \in ans(\tau : F)\}$
- Default interpretation $\mathrm{DI}(\tau : F)$: $\mathrm{DI}(\tau : F) \models p$ iff $p \in ans^+(\tau : F)$
- Default consistency: $\mathrm{DI}(\tau : F) \models ans^-(\tau : F)$

  ... *negated formulas constrain the default interpretations.*

$$ans(\mathrm{T} : \neg\neg p) \; = \; \{\neg\neg p\}$$

$$\mathrm{DI}(\mathrm{T} : \neg\neg p) \; = \; \emptyset, \qquad ans^-(\mathrm{T} : \neg\neg p) \; = \; \{\neg\neg p\},$$

Default inconsistent: $\emptyset \not\models \neg\neg p$

Answer sets for nested expressions using pieces of information

Basic concepts:

- Positive answers: $ans^+(\tau : F) = \{p \mid p \in ans(\tau : F)\}$,
- Negative answers: $ans^-(\tau : F) = \{\neg H \mid \neg H \in ans(\tau : F)\}$
- Default interpretation $\mathrm{DI}(\tau : F)$: $\mathrm{DI}(\tau : F) \models p$ iff $p \in ans^+(\tau : F)$
- Default consistency: $\mathrm{DI}(\tau : F) \models ans^-(\tau : F)$
  ... *negated formulas constrain the default interpretations.*
- I-stability: $\mathrm{DI}(\tau' : F) \subset \mathrm{DI}(\tau : F) \;\Rightarrow\; \mathrm{DI}(\tau : F) \not\models ans^-(\tau' : F)$

$$ans([1, \mathrm{T}] : p \vee \neg\neg p) \;=\; \{\, p \,\}, \quad ans([2, \mathrm{T}] : p \vee \neg\neg p) \;=\; \{\, \neg\neg p \,\},$$

$$\mathrm{DI}([1, \mathrm{T}] : p \vee \neg\neg p) \;=\; \{p\} \quad ans^-([1, \mathrm{T}] : p \vee \neg\neg p) \;=\; \emptyset,$$
$$\mathrm{DI}([2, \mathrm{T}] : p \vee \neg\neg p) \;=\; \emptyset \quad ans^-([2, \mathrm{T}] : p \vee \neg\neg p) \;=\; \{\neg\neg p\},$$

Instability: $\{p\} \models \neg\neg p$.

Answer sets for nested expressions using pieces of information

Basic concepts:

- Positive answers: $ans^{+}(\tau : F) = \{p \mid p \in ans(\tau : F)\}$,
- Negative answers: $ans^{-}(\tau : F) = \{\neg H \mid \neg H \in ans(\tau : F)\}$
- Default interpretation $\textsc{di}(\tau : F)$: $\textsc{di}(\tau : F) \models p$ iff $p \in ans^{+}(\tau : F)$
- Default consistency: $\textsc{di}(\tau : F) \models ans^{-}(\tau : F)$
  ... *negated formulas constrain the default interpretations.*
- I-stability: $\textsc{di}(\tau' : F) \subset \textsc{di}(\tau : F) \quad \Rightarrow \quad \textsc{di}(\tau : F) \not\models ans^{-}(\tau' : F)$

Definition (I-answer sets of nested expressions)

X is an I-answer set of F iff there is a default consistent and I-stable $\tau : F$ such that $X = \textsc{di}(\tau : F)$.

Equivalence with the standard definition of answer set

Theorem (Equivalence Theorem)

X is an I-answer set of F iff it is an answer set of F.

Example

$\tau_i : (p \vee q) \wedge \neg(p \wedge \neg q)$	$\mathrm{DI}(\tau : F)$	$ans^-(\tau : F)$
$\tau_1 = [\,[1, \mathrm{T}], \mathrm{T}\,]$	$\{p\}$	$\{\,\neg(p \wedge \neg q)\,\}$
$\tau_2 = [\,[2, \mathrm{T}], \mathrm{T}\,]$	$\{q\}$	$\{\,\neg(p \wedge \neg q)\,\}$

τ_1 not default consistent ($\mathrm{DI}(\tau_1 : F) \not\models ans^-(\tau_1 : F)$)
τ_2 default consistent ($\mathrm{DI}(\tau_2 : F) \models ans^-(\tau_2 : F)$) and I-stable:

$\{q\}$ is the unique I-answer set.

Equivalence with the standard definition of answer set

Theorem (Equivalence Theorem)

X is an I-answer set of F iff it is an answer set of F.

Example

$\tau_i : p \vee \neg\neg p$	$\mathrm{DI}(\tau : F)$	$ans^-(\tau : F)$
$\tau_1 = [1, \mathrm{T}]$	$\{p\}$	$\emptyset$
$\tau_2 = [2, \mathrm{T}]$	$\emptyset$	$\{\neg\neg p\}$

τ_1 not I-stable: $\mathrm{DI}(\tau_2 : F) \subset \mathrm{DI}(\tau_1 : F)$ and $\mathrm{DI}(\tau_1 : F) \models ans^-(\tau_2 : F)$.
τ_2 not default consistent ($\emptyset \not\models \neg\neg p$)

No I-answer set .

Proving the Equivalence Theorem by maps preserving $ans(\tau : F)$

- Ans-preserving map: $f : F \longrightarrow G$ such that $ans(\tau : F) = ans(f(\tau) : G)$
- Lemma 1: If there is $f : \mathrm{IT}(F) \leftrightarrow \mathrm{IT}(G)$ s.t. f and f^{-1} are ans-preserving, then for every H, $F \wedge H$ and $G \wedge H$ have the same I-answer sets
- Lemma 2: There is such a bijective $f : F \leftrightarrow FN$, where FN is a disjunction of conjunctions of simple formulas.
- Equivalence Theorem: it sufficies to prove that I-answer sets and answer sets coincide for FN.

Outline

1. Representing the CooML snapshot semantics in DLV
2. *Fcl* semantics vs stable model semantics
3. Conclusion: and now?

Conclusion

- Work in progress: theoretical analysis related to a DLV implementation of CooML snapshot generation
- Next (theoretical) steps: extend our work to nested programs (possibly, with $\exists$)
    - idea: extend ans^+, DI, ans^- to the larger language and relate I-stability to the minimality condition in $SM[F]$
      (SM defined in: "Stable Models and Circumscription", Ferraris, Lee, Lifschitz - 2007)

And now?

- A different characterisation is interesting
- Is there a realizability semantics based on stable pieces of information (i.e., maps preserving stable $\tau : F$)?
  - Discussion: there is no such map $f : \mathrm{IT}(p) \to \mathrm{IT}(\neg\neg p)$, but $p \to \neg\neg p$ is valid in HT (Here and There).
    HT characterizes strong equivalence, but not "strong implication".
- Is there a reasonable (at least valid) calculus for such a semantics?
  - Discussion: $p, \neg\neg p \vdash \neg\neg p$ should not be provable