$\mathcal{ID}$-LOGIC

Credits

- Presentation: Me
- Work done mainly by:
  - Marc Denecker (K.U. Leuven)
  - Eugenia Ternovska (SFU, Vancouver)
- And also part of the KRR-group at K.U. Leuven:
  - Maurice Bruynooghe
  - Nikolay Pelov
  - Maarten Mariën
  - Johan Wittockx

ID-logic

What is ID-logic?

ID-logic is an extension of classical logic with a new primitive for representing inductive definitions

Some questions

- Why are we interested inductive definitions?
- Why do we want a new primitive for this?
- Why extend classical logic?

(Inductive) Definitions in Mathematics

- "Ordinary" definitions

  A square is defined as a rectangle with equal height and width.

- Monotone inductive definitions

  The transitive closure $TC(R)$ of a binary relation R is inductively defined by the following rules:

  - $\forall x, y$, if $(x, y) \in R$, then $(x, y) \in TC(R)$;
  - $\forall x, y$, if $\exists z$, $(x, y), (y, z) \in TC(R)$, then $(x, z) \in TC(R)$.

(Inductive) Definitions in Mathematics (2)

- Non-monotone inductive definitions over a well-founded order

  We define the relation $S \models \phi$ between structures S and formulas ϕ by the following induction (over length of formulas):

  - $S \models P(t_1, \ldots, t_n)$ if $(t_1^S, \ldots, t_n^S) \in P^S$;
  - $S \models \phi \land \psi$ if $S \models \phi$ and $S \models \psi$;
  - $S \models \phi \lor \psi$ if $S \models \phi$ or $S \models \psi$;
  - $S \models \exists x\ \phi$ if there exists $d \in Dom(S)$, $S \models \phi[x/d]$
  - $S \models \neg\phi$ if $S \not\models \phi$

- Iterated inductive definitions

Inductive definitions

In mathematics

- Inductive definitions occur quite often
- Offer constructive characterization of certain concepts
- Useful and well-understood way for representing certain kind of knowledge

Two diffferent kinds of mathematical knowledge

- Definitions: State what a concept is
  - Inductive definitions are important part of this
- Propositions, theorems, ...: State properties of concepts

Classical logic

As a language for representing mathematical knowledge?

- Is well-suited for representing theorems etc.
- But expressing definitions is harder
  - $\forall x \; Square(x) \Leftrightarrow Rectangle(x) \wedge Height(x) = Width(x)$
  - $\forall x, y \; R(x, y) \Rightarrow TC(x, y)$
    $\forall x, y \; (\exists z \; TC(x, z) \wedge TC(z, y)) \Rightarrow TC(x, y)$
    $$\forall P \; \begin{pmatrix} (\forall x, y \; R(x, y) \Rightarrow P(x, y)) \wedge \\ (\forall x, y \; (\exists z \; P(x, z) \wedge P(z, y)) \Rightarrow P(x, y) \end{pmatrix} \Rightarrow TC \subseteq P$$
  - For non-monotone definitions: Explicitly encode well-founded order

New inductive definition primitive

Offers a uniform and straightforward way of representing all previously mentioned kinds of inductive definitions

In mathematical text

The transitive closure $TC(R)$ of a binary relation R is inductively defined by the following rules:

- $\forall x, y$, if $(x, y) \in R$, then $(x, y) \in TC(R)$;
- $\forall x, y$, if $\exists z, (x, y), (y, z) \in TC(R)$, then $(x, z) \in TC(R)$.

In ID-logic

$$\left\{ \begin{array}{l} \forall x, y \;\; TC(x, y) \leftarrow R(x, y). \\ \forall x, y \;\; TC(x, y) \leftarrow \exists z \;\; TC(x, z) \wedge TC(z, y). \end{array} \right\}$$

A definition in ID-logic

$$\left\{\begin{array}{l}\forall x, y \; TC(x,y) \leftarrow R(x,y). \\ \forall x, y \; TC(x,y) \leftarrow \exists z \; TC(x,z) \wedge TC(z,y).\end{array}\right\}$$

- Definition is a set of definitional rules between $\{\}$

$$\forall \mathbf{x} \; P(\mathbf{t}) \leftarrow \phi.$$

- Defines number of predicates
- Other predicates are open, i.e., supposed to be given

Semantics of a monotone definition Δ

Given an interpretation for the open predicates, the defined predicates should be interpreted by the least fixpoint of the operator that derives the heads of all ground instantiations of rules of Δ whose bodies are satisfied.

- Fix domain D and pre-interpretation F
- Let R be an interpretation in D of the open predicates of Δ
- Let T^R_Δ be the operator on interpretations in D of the defined predicates of Δ, that maps S to S', with for each defined P/n
  - $P^{S'} = \{\mathbf{d} \in D^n \mid$ for which
  - there exists $\forall \mathbf{x}\ P(\mathbf{t}) \leftarrow \phi. \in \Delta$ and $\mathbf{d_x} \in D^{|\mathbf{x}|}$ such that
  - $\mathbf{d} = \mathbf{t}^{F[\mathbf{x}/\mathbf{d_x}]}$ and $(R \cup S)[\mathbf{x}/\mathbf{d_x}] \models \phi\}$
- Given this interpretation R for the open predicates, the interpretation of the defined predicates should be $\mathrm{lfp}(T^R_\Delta)$

How to extend this to non-monotone definitions?

Semantics of such a definition (2)

Well-founded model

- Generalizes the intuitions behind least fixpoint construction
- To also apply to non-monotone definitions

For the kind of definitions typically occuring in mathematics

- This semantics coincides with their "usual interpretation"

For more complicated definitions

- It also does something reasonable

Formal semantics of a definition

$S \models \Delta$ iff

- Use S to interpret open predicates of Δ
- Construct $wfm_{S|_{Open_\Delta}}(\Delta)$ of Δ given $S|_{Open_\Delta}$
- $wfm_{S|_{Open_\Delta}}(\Delta)$ is two-valued and equal to $S|_{Def_\Delta}$

Note that, although the semantics uses WFS which can be three-valued, models of a definitions are two-valued

Formal semantics: Example

$$\left\{\begin{array}{l}\forall x, y \;\; TC(x,y) \leftarrow R(x,y). \\ \forall x, y \;\; TC(x,y) \leftarrow \exists z \;\; TC(x,z) \wedge TC(z,y).\end{array}\right\}$$

$$\left\{\begin{array}{c} R(A,B), R(B,C), \\ TC(A,B), TC(B,C), TC(A,C) \end{array}\right\} \models \Delta_{TC}$$

- If we interpret $R/2$ by $\{R(A,B), R(B,C)\}$
- The wfm of Δ_{TC} is $(\{TC(A,B), TC(B,C), TC(A,C)\},$
  $\{TC(A,B), TC(B,C), TC(A,C)\})$

ID-logic in full

Syntax is inductively defined as:

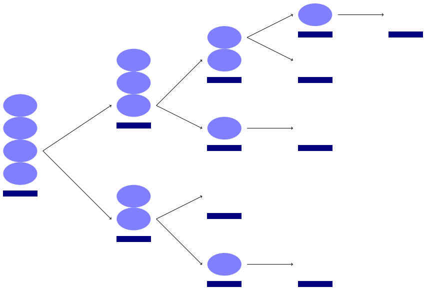
- A definition Δ is a formula
- An atom $P(\mathbf{t})$ is a formula
- Disjunctions, conjunctions, negations, quantifications of formulas are also formulas

Semantics is inductively defined as:

- $S \models \Delta$ if $wfm_{S|_{Open_\Delta}}(\Delta)$ is two-valued and equal to $S|_{Def_\Delta}$
- $S \models P(\mathbf{t})$ if $\mathbf{t}^S \in P^S$
- Usual induction for connectives

A theory T is a set of formulas and $S \models T$ iff $\forall \phi \in T, S \models \phi$

Example: Winning positions of a game



The game:

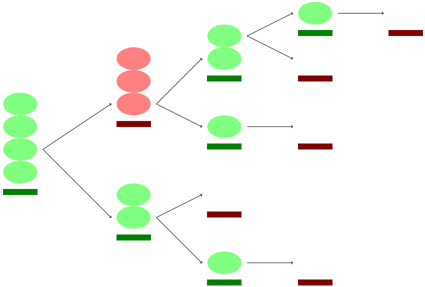
- Two players
- Stack of N stones
- Remove 1 or 2 stones
- Last move wins

$$\forall x \; Pos(x) \Leftrightarrow (x \leq N) \wedge (x \geq 0)$$

$$\left\{ \begin{array}{l} \forall x, y \; Move(x,y) \leftarrow Pos(x) \wedge Pos(y) \\ \qquad\qquad\qquad\qquad \wedge (y = x - 1 \vee y = x - 2). \end{array} \right\}$$

$$\{\forall x \; Win(x) \leftarrow \exists y \; Move(x,y) \wedge \neg Win(y).\}$$

Example: Winning positions of a game



The game:

- Two players
- Stack of N stones
- Remove 1 or 2 stones
- Last move wins

$$\forall x \; Pos(x) \Leftrightarrow (x \leq N) \wedge (x \geq 0)$$

$$\left\{ \begin{aligned} &\forall x, y \;\; Move(x, y) \leftarrow Pos(x) \wedge Pos(y) \\ &\qquad\qquad\qquad\qquad \wedge (y = x - 1 \vee y = x - 2). \end{aligned} \right\}$$

$$\{\forall x \;\; Win(x) \leftarrow \exists y \;\; Move(x, y) \wedge \neg Win(y).\}$$

Behaviour of definitions

Two different types of updates

- Adding rule to definition
  - Non-monotone
  - The set of formulas that hold in all models might decrease
- Adding a definition
  - Monotone
  - The set of formulas that hold in all models increases
  - Same as adding a FOL-formula

ID-logic adds a restricted form of non-monotonicity, while retaining the overall monotonicity

Usefulness

- An inductive definitions is mathematical concept
- But strongly rooted in intuition
  - Describes a constructive process, that is very natural
  - Similar to human thought process?
  - Similar to cause-effect propagations in the real world?
- ID-logic is also a useful tool for
  - Formalizing (aspects of) common sense reasoning
  - Solving computational problems

Defaults

General Principle

If you define a relation, then tuples do not belong to this relation unless there is an explicit reason for them to do so

- $\{\forall x\ \mathit{Flies}(x) \leftarrow \mathit{Bird}(x) \wedge \neg \mathit{Ab}(x).\}$
- $\left\{ \begin{array}{r} \mathit{Bird}(\mathit{Tweety}). \\ \mathit{Bird}(\mathit{Fred}). \end{array} \right\} \wedge \{\mathit{Penguin}(\mathit{Fred}).\}$
- $\{\forall x\ \mathit{Ab}(x) \leftarrow \mathit{Penguin}(x).\}$

Ontologies

ID-logic is a very natural fit, e.g.,

$$\left\{\begin{array}{l} \forall x\ Mammal(x) \leftarrow Placental\_Mammal(x). \\ \forall x\ Mammal(x) \leftarrow Marsupil(x). \\ \forall x\ Mammal(x) \leftarrow Monotreme(x). \end{array}\right\}$$

$$\left\{\begin{array}{c} \forall x\ Placental\_Mammal(x) \leftarrow Primate(x). \\ \forall x\ Placental\_Mammal(x) \leftarrow Rodent(x). \\ \dots \end{array}\right\}$$

- Combine with defaults
- Multiple subdivisions

$$\left\{\begin{array}{l} \forall x\ Human(x) \leftarrow Man(x). \\ \forall x\ Human(x) \leftarrow Woman(x). \end{array}\right\} \wedge \left\{\begin{array}{l} \forall x\ Human(x) \leftarrow Child(x). \\ \forall x\ Human(x) \leftarrow Adult(x). \end{array}\right\}$$

Situation calculus: Gear wheel example

Fluents: for all g

- $Turn(g)$: Gear wheel g is turning

Actions: for all g

- $Start(g)$: Start gear wheel g
- $Stop(g)$: Start gear wheel g

For every fluent $f(\mathbf{x})$: $H_f(\mathbf{x}, S)$, $Init_f(\mathbf{x})$, $C_f(\mathbf{x}, A, S)$, $C_{\neg f}(\mathbf{x}, A, S)$

$$\left\{ \begin{array}{ll} \forall g & H_{Turn}(g, S_0) \leftarrow Init_{Turn}(g). \\ \forall g, a, s & H_{Turn}(g, do(a, s)) \leftarrow H_{Turn}(g, s) \wedge \neg C_{\neg Turn}(g, a, s). \\ \forall g, a, s & H_{Turn}(g, do(a, s)) \leftarrow C_{Turn}(g, a, s). \end{array} \right\}$$

Gear wheel example (2)

Domain specific part:

- (Direct and indirect) effects of actions

$$\left\{\begin{array}{lr} \forall g, s & C_{Turn}(g, Start(g), s). \\ \forall g, s & C_{\neg Turn}(g, Stop(g), s). \\ \forall g, s & C_{Turn}(g, a, s) \leftarrow \exists h\ C_{Turn}(h, a, s) \wedge Conn(h, g). \\ \forall g, s & C_{\neg Turn}(g, a, s) \leftarrow \exists h\ C_{\neg Turn}(h, a, s) \wedge Conn(h, g). \end{array}\right\}$$

- Particular instance:

$$\left\{\begin{array}{lr} & Conn(1, 2). \\ \forall x, y & Conn(x, y) \leftarrow Conn(y, x). \end{array}\right\}$$

- Initial situation: $\neg\exists g\ Initially_{Turn}(g)$

Hamiltonian Path

$Ham(x, y)$ means that the edge (x, y) is in the path

$$\left\{ \begin{array}{l} \forall x\ Reached(x) \leftarrow \exists y\ Initial(y) \wedge Ham(y, x). \\ \forall x\ Reached(x) \leftarrow \exists y\ Reached(y) \wedge Ham(y, x). \end{array} \right\}$$

$$\forall x, y, z\ Ham(x, y) \wedge Ham(x, z) \Rightarrow z = y$$

$$\forall x, y, z\ Ham(x, y) \wedge Ham(z, y) \Rightarrow z = x$$

$$\forall x, y\ Ham(x, y) \Rightarrow Edge(x, y)$$

$$\forall x\ Vertex(x) \Rightarrow Reached(x)$$

- Input: $\{Vertex(A).\ \cdots\ Vertex(S).\}\ \{Initial(A).\}$
  $\{Edge(A, B).\ \cdots\ Edge(R, S).\}$
- Find paths by generating Herbrand models

Different approach

A Framework for Representing and Solving NP Search Problems. *D. Mitchell and E. Ternovska.* *Proc. AAAI 2005.*

- Use same theory
- Input: let R be a structure for $\{Vertex/1, Edge/2, Initial/1\}$
  - $Domain(R) = \{A, B, \ldots\}$
  - $Initial^R = \{A\}$
  - $Vertex^R = Domain(R)$
  - $Edge^R = \{(A, B), \ldots\}$
- Find paths by extending R to model for theory
- Complexity result: MX(FO(ID)) captures NP

Implementation: MidL

Main idea

- ID-logic extends classical logic with inductive definitions
- MidL extends SAT-solving with incremental WFM computation
  - Guess open atom
  - Propagate by SAT clauses
  - Propogate by definitions

Focus

- Achieving good integration between these two components
- E.g. watched literals, clause learning,...

Download:

`http://www.cs.kuleuven.be/∼maartenm/research/midl.html`

The relation with ASP

- Informally: very different
  - ID-logic is about representing inductive definitions
  - ASP is about representing rules governing the beliefs of a rational agent
- Formally: very similar
- Practical examples: very similar
- Straightforward transformations between large subsets of ID-logic and extended LPs
- The constructive process described by an inductive definition corresponds to reasoning process of a rational agent?

Main difference

ID-logic has no epistemic component, i.e., things like $check : -not\ orphan, not\ \neg orphan.$ are not possible

Some correspondences

- Definitions
  - $$\left\{\begin{array}{l} Reached(x) \leftarrow \exists y\ Reached(y) \land Ham(y,x). \\ Reached(x) \leftarrow Initial(x). \end{array}\right\}$$
  - $$\begin{array}{r} reached(X) :- reached(Y), ham(Y,X). \\ reached(X) :- initial(X). \\ \neg reached(X) :- not\ reached(X). \end{array}$$
- Open predicates (i.e., not defined anywhere)
  - Nothing in ID-logic
  - $ham(X,Y) \lor \neg ham(X,Y).$
  - or $$\begin{array}{r} ham(X,Y) :- not\ \neg ham(X,Y). \\ \neg ham(X,Y) :- not\ ham(X,Y). \end{array}$$
- Assertions
  - $\forall x,y,z\ Ham(x,y) \land Ham(x,z) \Rightarrow z = y$
  - $:- ham(X,Y), ham(X,Z), Z <> Y.$

Contributions of ID-logic to LP

- LPs can be used to represent inductive definitions
  - Well known for monotone definitions
  - WFS extends this to non-monotone definitions
- ID-logic isolates this informal "fragment" of LP
  - Clear, unambiguous and well-known intuitive reading
- Shows how it can be integrated into classical logic
  - By adding a single new primitive to the language
- Allows it to be used together with all the expertise for classical logic

ID-logic shows

One clearcut contribution that LP has to offer to classical logic is its ability to represent inductive definitions

Introduction to ID-logic

ID-logic and common sense
- Defaults
- Ontologies
- Actions

ID-logic as a computational tool

The relation to ASP and LP

Future work & Conclusion

Work being done on ID-logic

- Most of the research in our group is devoted/related to ID-logic
  - Develop model generator
  - Develop a proof system (preliminary)
  - Study relation with fixpoint logics
  - Use ID-logic for data integration
- At SFU:
  - Study MX(FO(ID))
  - Using ID-logic for verification
- My interests
  - Proving some modularity results
  - Relation between ID-logic and CP-logic

Relation between inductive definitions and causality

- An inductive definition implicitely describes a mathematical object as the outcome of a derivation process governed by a set of rules
- A CP-theory implicitely describes a probability distribution as the outcome of a derivation process governed by a set of conditional probabilistic experiments
- The intuitions (and mathematics) seem very closely related
- Is an inductive definition simply a set of deterministic causal rules in the context of mathematical objects?
- What if we replace the "inductive definition"-primitive by a "probabilistic causal process"-primitive?

Conclusion

- ID-logic introduces a new primitive that allows inductive definitions to be represented in a uniform and straightforward way
- Inductive definitions are important
  - In mathematics
  - For common sense reasoning
  - For solving computational problems
- ID-logic integrates this in classical logic
  - Reuse all old expertise
- Everyone who knows classical logic and who can read an inductive definition, already knows ID-logic

http://www.cs.kuleuven.be/∼maartenm/research/midl.html