Abductive Reasoning in Intuitionistic Propositional Logic via Theorem Synthesis

Paul Tarau

University of North Texas

AUGUST 2, 2022

ICLP'2022

Overview

- we synthesize minimal assumptions under which a given formula in Intuitionistic Propositional Logic (**IL**) becomes a theorem
- our tool: a compact Prolog-based **IL** theorem prover
- $\Rightarrow$ an abductive reasoning mechanism for **IL**
- a generalization of abduction: we synthesize sequent premises using a set of canonical formulas covering via a reduction mechanism arbitrary **IL** formulas
- the paper is a self-contained literate Prolog program, with code at: `https://github.com/ptarau/TypesAndProofs/blob/master/isynt.pro`.

The Truth Tables of Classical Logic (**CL**)

- given a formula F in **CL**, each row in a formula's truth table describes a conjunction of literals C
- as an example, let us consider the **CL** formula
  `F = (A v B) & (B v C) & (C v A).`

```
 A B C :  F                                                A B C:  G->F
[0,0,0]-->0                                               [0,0,0]-->1
[0,0,1]-->0                                               [0,0,1]-->1
[0,1,0]-->0                                               [0,1,0]-->1
[0,1,1]-->1                                               [0,1,1]-->1
[1,0,0]-->0                                               [1,0,0]-->1
[1,0,1]-->1  <= selected row                              [1,0,1]-->1
[1,1,0]-->1                                               [1,1,0]-->1
[1,1,1]-->1                                               [1,1,1]-->1
```

- select a row, say [1,0,1]–>1 and interpret it as `G = A & ~B & C`
- reading the truth table as a disjunctive normal form, it immediately follows that $C \rightarrow F$ is a tautology
- the truth table of the resulting tautology `G -> F` is shown in the right column.

The Envy for the Classics: Can we replicate this in **IL**?

- contrary to **CL** and intermediate logic, in **IL** we have:
  - *no finite truth-tables*, no inter-definability of logical connectives
  - no rule of excluded middle
  - only a concept of *tautology* and *contradiction*
- $\Rightarrow$ in **IL** we need to find assumptions that would make the formula a theorem
- such assumptions include conjunctions of literals, mimicking the truth tables of **CL**, but it also makes sense to extend them to more expressive subsets of formulas
- given a formula in **IL**, we will need a search process for finding assumptions that would make it a theorem
- however, we would like our assumptions to be minimal with respect to the partial order relation governing the logic: *intuitionistic implication*

An Inspiration Source: Abductive Logic Programming

- Abductive **LP**: facts designated as *abducibles* are filtered with integrity constraints to provide relevant assumptions needed for the success of a goal G w.r.t. a given program P
- in the context of **IL**, our abductive reasoning algorithm will rely on finding *minimal assumptions under which a formula becomes a theorem*
- in the absence of a convenient automated semantic method like truth tables or SAT solvers in **CL**, we will need a theorem prover, ideally derived directly from the rules of a *terminating* sequent calculus

Roy Dyckhoff's **G4ip** sequent calculus

termination ensured with "multiset ordering", no loop checking is needed!

$$\Gamma, p \Rightarrow p \quad Ax \quad (p \text{ an atom})$$

$$\Gamma, \bot \Rightarrow \Delta \quad L\bot$$

$$\frac{\Gamma \Rightarrow \varphi \quad \Gamma \Rightarrow \psi}{\Gamma \Rightarrow \varphi \wedge \psi}\ R\wedge$$

$$\frac{\Gamma, \varphi, \psi \Rightarrow \Delta}{\Gamma, \varphi \wedge \psi \Rightarrow \Delta}\ L\wedge$$

$$\frac{\Gamma \Rightarrow \varphi_i}{\Gamma \Rightarrow \varphi_0 \vee \varphi_1}\ R\vee \ (i = 0, 1)$$

$$\frac{\Gamma, \varphi \Rightarrow \Delta \quad \Gamma, \psi \Rightarrow \Delta}{\Gamma, \varphi \vee \psi \Rightarrow \Delta}\ L\vee$$

$$\frac{\Gamma, \varphi \Rightarrow \psi}{\Gamma \Rightarrow \varphi \to \psi}\ R\to$$

$$\frac{\Gamma, p, \varphi \Rightarrow \Delta}{\Gamma, p, p \to \varphi \Rightarrow \Delta}\ Lp\to \ (p \text{ an atom})$$

$$\frac{\Gamma, \varphi \to (\psi \to \gamma) \Rightarrow \Delta}{\Gamma, \varphi \wedge \psi \to \gamma \Rightarrow \Delta}\ L\wedge\to$$

$$\frac{\Gamma, \varphi \to \gamma, \psi \to \gamma \Rightarrow \Delta}{\Gamma, \varphi \vee \psi \to \gamma \Rightarrow \Delta}\ L\vee\to$$

$$\frac{\Gamma, \psi \to \gamma \Rightarrow \varphi \to \psi \quad \gamma, \Gamma \Rightarrow \Delta}{\Gamma, (\varphi \to \psi) \to \gamma \Rightarrow \Delta}\ L\to\to$$

The Theorem Prover for **IL**

- we derive the prover directly from Roy Dyckhoff's G4ip calculus

```
iprover(true,_):-!.
iprover(A,Vs):-memberchk(A,Vs),!.
iprover(_,Vs):-memberchk(false,Vs),!.
iprover(~A,Vs):-!,iprover(false,[A|Vs]).
iprover(A<->B,Vs):-!,iprover(B,[A|Vs]),iprover(A,[B|Vs]).
iprover((A->B),Vs):-!,iprover(B,[A|Vs]).
iprover((B<-A),Vs):-!,iprover(B,[A|Vs]).
iprover(A & B,Vs):-!,iprover(A,Vs),iprover(B,Vs).
iprover(G,Vs1):- % atomic or disj or false
  select(Red,Vs1,Vs2),
  iprover_reduce(Red,G,Vs2,Vs3),
  !,
  iprover(G,Vs3).
iprover(A v B, Vs):-(iprover(A,Vs) ; iprover(B,Vs)),!.
```

The Prover, continued

- we delegate details to helper predicates: `iprover_reduce/4` and `iprover_impl/4`.

```
iprover_reduce(true,_,Vs1,Vs2):-!,iprover_impl(false,false,Vs1,Vs2).
iprover_reduce(~A,_,Vs1,Vs2):-!,iprover_impl(A,false,Vs1,Vs2).
iprover_reduce((A->B),_,Vs1,Vs2):-!,iprover_impl(A,B,Vs1,Vs2).
iprover_reduce((B<-A),_,Vs1,Vs2):-!,iprover_impl(A,B,Vs1,Vs2).
iprover_reduce((A & B),_,Vs,[A,B|Vs]):-!.
iprover_reduce((A<->B),_,Vs,[(A->B),(B->A)|Vs]):-!.
iprover_reduce((A v B),G,Vs,[B|Vs]):-iprover(G,[A|Vs]).
```

```
iprover_impl(true,B,Vs,[B|Vs]):-!.
iprover_impl(~C,B,Vs,[B|Vs]):-!,iprover((C->false),Vs).
iprover_impl((C->D),B,Vs,[B|Vs]):-!,iprover((C->D),[(D->B)|Vs]).
iprover_impl((D<-C),B,Vs,[B|Vs]):-!,iprover((C->D),[(D->B)|Vs]).
iprover_impl((C & D),B,Vs,[(C->(D->B))|Vs]):-!.
iprover_impl((C v D),B,Vs,[(C->B),(D->B)|Vs]):-!.
iprover_impl((C<->D),B,Vs,[((C->D)->((D->C)->B))|Vs]):-!.
iprover_impl(A,B,Vs,[B|Vs]):-memberchk(A,Vs).
```

- Classical Logic “for free”, via Glivenko’s theorem:

```
cprover(T):-iprover( ~ ~T).
```

Abductive Reasoning Mechanisms

- defining some of the atoms occurring in a formula F as the only ones to be used in the search process brings us to declare them as *abducibles*
- *Protasis generation*: assume of a subset of abducibles and their negations

```
any_protasis(Prover,AggregatorOp,WithNeg,Abducibles,Formula,Assumption):-
  abducibles_of(Formula,Abducibles),
  mark_hypos(WithNeg,Abducibles,Literals),
  subset_of(Literals,Hypos),
  join_with(AggregatorOp,Hypos,Assumption),
  \+ (call(Prover,Assumption->false)), % we do not assume contradictions !
  call(Prover,Assumption->Formula).    % we ensure this is a theorem
```

- an `AggregatorOp` that mimicsTruth Tables, is conjunction "`&`"

Finding the Weakest Protasis

- a *partial order* in **IL** is defined by the intuitionistic implication “->”
- it is a total order in **CL** where `(p->q) v (q->p)` is a theorem
- for Peirce’s law `((p->q)->p)->p` (a theorem in **CL**), we obtain a weakest protasis: `p v ~p` which would indeed turn **IL** into **CL**
- the *logic of here-and-there* is derived from **IL** by adding the axiom
  `f v (f->g) v ~g`
- the weakest protasis indicates that the excluded middle rule would need to hold for `f` or for `g`:

```
?- weakest_protasis(iprover,(v),yes,_,(f v (f->g) v ~g),P).
P = f v ~f v g v ~g.
```

An Example of Intuitionistic Abductive Reasoning

- explaining with a Prover, by setting Integrity Constraints:

```
explain_with(Prover,Abducibles,Prog,IC,G,Expl):-
    any_protasis(Prover, (&),yes,Abducibles, (Prog->G), Expl),
    call(Prover, Expl & Prog->G),    % ensure it we explain the goal
    call(Prover, (Expl & Prog->IC)), % ensure the integrity constraints hold
    \+ (call(Prover, (Expl & Prog -> false))). % Expl are consistent with Prog
```

- the program:

```
why_wet(Prover):-
    IC = ~(rained & sunny),
    P = sunny & (rained v sprinkler -> wet), As=[sprinkler,rained], G = wet,
    writeln(prog=P), writeln(ic=IC),
    explain_with(Prover,As,P,IC,G,Explanation),
    writeln('Explanation:' --> Explanation).
```

- the query:

```
?- why_wet(iprover).
prog=sunny&(rained v sprinkler->wet)
ic= ~ (rained&sunny)
Explanation: --> sprinkler& ~rained
```

Minimal Canonical Assumptions: the Mints Transformation

- Grigori Mints has proven that any formula f is equiprovable to a formula of the form $X_f \rightarrow g$, where X_f is a conjunction of formulas of one of the forms:

$$p, \tilde{}p,\ p \rightarrow q,\ (p \rightarrow q) \rightarrow r,\ p \rightarrow (q \rightarrow r),\ p \rightarrow (q\ v\ r),\ p \rightarrow \tilde{}q,\ \tilde{}q \rightarrow p.$$

- $\Rightarrow$ we have a set of small canonical assumptions that can to turn a given formula into a theorem in **IL**!
- a "weakest Mints premise" is defined along the lines of the weakest protasis
- example:

```
?- weakest_mints_premise(iprover,_,(f v (f->g) v ~g),P).
P = f ; P = ~g ; P = ( f->g) ; P = ( g-> ~g).
```

Discussion

- Abductive LP, ASP, (s)CASP share as a key idea model synthesis
- models are sets of facts that hold under assumptions described by a program
- ILP aims to synthesize rules that describe sets of facts more compactly
- our aim using the Mints canonical form for assumptions is somewhere in-between
- given that an equiprovable sequent to a given formula can be built from a set of Mints formulas, our synthesis algorithm stays as expressive while being restricted to them as if arbitrary formulas would be considered

Conclusion

- we provide a generalized view of abductive reasoning as an instance of program synthesis controlled by a theorem prover
- this approach can be applied to interesting intermediate logics among which the equilibrium-logic (relevant as a foundation of **ASP** systems) as well as modal logics and their instantiations as alethic, temporal, deontic or epistemic systems
- besides providing (in the form of the concept of weakest protasis) an analogue of the unavailable truth-table models for intuitionistic formulas, we have also generalized our abduced sequent premises to use minimal canonical formulas to which arbitrary **IL** formulas can be broken down, with the potential of synthesizing a richer set of "salient" assumptions that would make a given formula a theorem
- this generalized abduction synthesis could reveal critical missing assumptions, not just as literals but also as a conjunction of *interdependencies* among them