Abductive Reasoning in Intuitionistic Propositional Logic via Theorem Synthesis

Paul Tarau

University of North Texas

AUGUST 2, 2022

ICLP’2022
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

⇒ 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.
given a formula $F$ in $\textbf{CL}$, each row in a formula’s truth table describes a conjunction of literals $C$.

as an example, let us consider the $\textbf{CL}$ formula $F = (A \lor B) \land (B \lor C) \land (C \lor A)$.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

select a row, say $[1,0,1] \rightarrow 1$ and interpret it as $G = A \land \neg B \land C$

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 $G \rightarrow F$ is shown in the right column.
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

⇒ 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*
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!

<table>
<thead>
<tr>
<th>Rule</th>
<th>Premise</th>
<th>Conclusion</th>
</tr>
</thead>
</table>
| Ax   | \( \Gamma, p \Rightarrow p \) \hspace{1cm} (p \text{ an atom}) | \(
| R\land | \Gamma \Rightarrow \phi \quad \Gamma \Rightarrow \psi | \Gamma \Rightarrow \phi \land \psi |
| R\lor | \Gamma \Rightarrow \phi_i | \Gamma \Rightarrow \phi_0 \lor \phi_1 \hspace{1cm} (i = 0, 1) |
| R\rightarrow | \Gamma, \phi \Rightarrow \psi | \Gamma \Rightarrow \phi \rightarrow \psi |
| L\land | \Gamma, \phi \Rightarrow \psi \quad \Gamma, \psi \Rightarrow \Delta | \Gamma \Rightarrow \phi \rightarrow \psi \land \psi \Rightarrow \Delta |
| L\lor | \Gamma, \phi \rightarrow \psi \land \psi \Rightarrow \Delta | \Delta \Rightarrow \Gamma \land \psi \rightarrow \psi \land \psi \Rightarrow \Delta |
| L\rightarrow | \Gamma, \phi \rightarrow \psi \land \psi \Rightarrow \Delta | \Delta \Rightarrow \Gamma \land \psi \rightarrow \psi \land \psi \Rightarrow \Delta |
| L\rightarrow | \Gamma, \phi \rightarrow \psi \land \psi \Rightarrow \Delta | \Delta \Rightarrow \Gamma \land \psi \rightarrow \psi \land \psi \Rightarrow \Delta |

Paul Tarau  
University of North Texas  
Abductive Reasoning via Theorem Synthesis  
AUGUST 2, 2022 6 / 14
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)), !.
```
we delegate details to helper predicates: \texttt{iprover\_reduce/4} and \texttt{iprover\_impl/4}.

\begin{verbatim}
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\rightarrow B),_,Vs1,Vs2):-!,iprover\_impl(A,B,Vs1,Vs2).
iprover\_reduce((B\leftarrow A),_,Vs1,Vs2):-!,iprover\_impl(A,B,Vs1,Vs2).
iprover\_reduce((A & B),_,Vs,[A,B|Vs]):-!.
iprover\_reduce((A\leftrightarrow B),_,Vs,[A\rightarrow B],[B\rightarrow 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\rightarrow false),Vs).
iprover\_impl((C\rightarrow D),B,Vs,[D\rightarrow B]|Vs)):--!.
iprover\_impl((D\leftarrow C),B,Vs,[D\rightarrow B]|Vs)):--!.
iprover\_impl((C & D),B,Vs,[(C\rightarrow (D\rightarrow B))|Vs]):--!.
iprover\_impl((C v D),B,Vs,[(C\rightarrow (D\rightarrow B))|Vs]):--!.
iprover\_impl((C\leftrightarrow D),B,Vs,[(C\rightarrow D)\rightarrow ((D\rightarrow C)\rightarrow B)|Vs]):--!.
iprover\_impl(A,B,Vs,[B|Vs]):-memberchk(A,Vs).
\end{verbatim}

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

\begin{verbatim}
cprover(T):-iprover(~ ~T).
\end{verbatim}
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

```prolog
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 mimics Truth Tables, is conjunction “&”
a *partial order* in IL is defined by the intuitionistic implication “$\rightarrow$”

it is a total order in CL where $(p \rightarrow q) \lor (q \rightarrow p)$ is a theorem

for Peirce’s law $( (p \rightarrow q) \rightarrow p ) \rightarrow p$ (a theorem in CL), we obtain a weakest protasis: $p \lor \neg p$ which would indeed turn IL into CL

the *logic of here-and-there* is derived from IL by adding the axiom

$$f \lor (f \rightarrow g) \lor \neg g$$

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:

```prolog
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:

```prolog
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:

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

$$p, \neg p, p \rightarrow q, (p \rightarrow q) \rightarrow r, p \rightarrow (q \rightarrow r), p \rightarrow (q \lor r), p \rightarrow \neg q, \neg q \rightarrow p.$$ 

⇒ 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).

$P = f$ ; $P = \neg g$ ; $P = (f->g)$ ; $P = (g->\neg g)$. 
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
we provide a generalized view of abductive reasoning as an instance of program synthesis controlled by a theorem prover

d 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