Analytic Proof Calculi

for the Notion of Failure

Joint Logic Workshop: Logic in Computer Science and Deduction Systems

Dov M. Gabbay and Ross Horne

Computer Science, University of Luxembourg

26 March 2021

Aim

We show how to add negation as failure

$$\neg P$$

to (a fragment of)

intuitionistic logic

in a methodological generic way.

Aim

We show how to add negation as failure

$$\neg P$$

to (a fragment of)

intuitionistic logic

in a methodological generic way.

¬c-- > a

(c -- > a)--> c

Short-term aim: Tame the failure of deduction (cut elimination) in N-prolog: we design an **analytic proof calculus**.

N-prolog = Intuitionistic implication truth, falsity and negation as failure, computed like prolog.

Long-term aim: Propose a method that can be applied to other logics, such as *linear logics* useful for internalising **resources** and **processes**.

Semantic point of view

Consider **intuitionistic proposition logic**, with distinct atoms Q, and connectives

$$\{\to, \bot, \top\}$$

[1] Observation adapted by Schroeder-Heister to handle inequality.

Semantic point of view

Consider **intuitionistic proposition logic**, with distinct atoms Q, and connectives

$$\{\rightarrow, \bot, \top\}$$

This logic has a canonical Kripke model $(S, \leq)$ where:

- S is all wffs of the language.
- $A \leq B$ iff $B \vdash_{Int} A$.
- $A \vDash b$ iff $A \vdash b$, where $b \in Q$ is atomic.
- $A \vDash B_1 \rightarrow B_2$ iff for any $A' \geq A$, if $A' \vDash B_1$ then $A' \vDash B_2$.
- $A \vDash \top$
- $A \nvDash \bot$.

[1] Observation adapted by Schroeder-Heister to handle inequality.

Semantic point of view

Consider **intuitionistic proposition logic**, with distinct atoms Q, and connectives

$$\{\rightarrow, \bot, \top\}$$

This logic has a canonical Kripke model $(S, \leq)$ where:

- S is all wffs of the language.
- $A \leq B$ iff $B \vdash_{Int} A$.
- $A \vDash b$ iff $A \vdash b$, where $b \in Q$ is atomic.
- $A \vDash B_1 \rightarrow B_2$ iff for any $A' \geq A$, if $A' \vDash B_1$ then $A' \vDash B_2$.
- $A \vDash \top$
- $A \nvDash \bot$.

In the cannonical model we have: $A \vDash B$ iff $A \vdash_{Int} B$, for any A, B.

So, $A \nvdash_{Int} B_1 \rightarrow B_2$ only if
for some A', $A' \vdash_{Int} A$ and $A' \vdash_{Int} B_1$ and $A' \nvdash_{Int} B_2$.[1]

Not analytic since A' need not be a subformula of A or $B_1 \rightarrow B_2$.

[1] Observation adapted by Schroeder-Heister to handle inequality.

Our idea

We add **Gödel-Tarski provability constants**:

For any Δ, A add the constant $\mathrm{Pr}(\Delta, A)$, where Δ is a finite set of wffs and A is a wff.

We let:

$$\mathrm{Pr}(\Delta, A) = \begin{cases} \top & \text{if } \Delta \vdash_{Int} A \\ \bot & \text{if } \Delta \nvdash_{Int} A \end{cases}$$

Let X be a variable ranging over finite sets of wffs. Then $\mathrm{Pr}(X, A)$ does not have a value unless X is instantiated.

By playing with how we instantiate X we can capture a variety of problems (answer set programming, abduction problems, preferred extensions, etc.). E.g., **scoped negation as failure**, i.e., failure from a specific database X becomes

$$\mathrm{Pr}(X, A) \to \bot$$

For example the following holds, if $X \vdash b$.

$$\vDash (\mathrm{Pr}(X, b) \to \bot) \to a$$

First design step: a proof calculus of success ...and failure

Calculus of success

$$\frac{}{\Delta, q \vdash q}\,(atom) \qquad \frac{}{\Delta, \bot \vdash A}\,(\bot) \qquad \frac{}{\Delta \vdash \top}\,(\top)$$

$$\frac{\Delta \vdash A \qquad x = q \text{ or } x = \bot}{\Delta, A \to x \vdash q}\,(elim) \qquad \frac{\Delta, A \vdash q}{\Delta \vdash A \Rightarrow q}\,(intro)$$

First design step: a proof calculus of success ...and failure

Calculus of success

$$\frac{}{\Delta, q \vdash q}\,(atom) \qquad \frac{}{\Delta, \bot \vdash A}\,(\bot) \qquad \frac{}{\Delta \vdash \top}\,(\top)$$

$$\frac{\Delta \vdash A \qquad x = q \text{ or } x = \bot}{\Delta, A \to x \vdash q}\,(elim) \qquad \frac{\Delta, A \vdash q}{\Delta \vdash A \Rightarrow q}\,(intro)$$

Calculus of failure

$$\frac{\Delta, A \nvdash q}{\Delta \nvdash A \Rightarrow q}\,(\neg intro)$$

$$\frac{\left\{\Delta \nvdash A : \begin{array}{l}\text{for any } A \text{ such that} \\ (A \Rightarrow x) \in \Delta, \text{ where } x = q \text{ or } x = \bot\end{array}\right\} \quad q \notin \Delta \quad \bot \notin \Delta}{\Delta \nvdash q}\,(\neg atom)$$

$$\frac{\left\{\Delta \nvdash A : \quad \text{for } A \text{ such that } (A \Rightarrow \bot) \in \Delta\right\} \quad \bot \notin \Delta}{\Delta \nvdash \bot}\,(\neg\bot)$$

A digression: naïvely adding NAF means deduction fails

First attempt: Following N-Prolog directly we could try the following naïve rules for fibering the systems of success and failure together using NAF.

$$\frac{\Delta \nvdash a}{\Delta \vdash \neg a}\,(\mathit{to\ fail}) \qquad \frac{\Delta \vdash a}{\Delta \nvdash \neg a}\,(\mathit{to\ succ})$$

A digression: naïvely adding NAF means deduction fails

First attempt: Following N-Prolog directly we could try the following naïve rules for fibering the systems of success and failure together using NAF.

$$\frac{\Delta \nvdash a}{\Delta \vdash \neg a}\,(\textit{to fail}) \qquad \frac{\Delta \vdash a}{\Delta \nvdash \neg a}\,(\textit{to succ})$$

An issue: it fails cut elimination (hence modus ponens)

> *"There are reasons to think that the addition of negation-as-failure to intuitionistic logic programming will lead to serious semantic diffculties. Gabbay studied the problem in [N-Prolog 1985], and concluded that the entire idea was logically incoherent, since modus ponens would no longer be valid." Boner and McCarty 1990*

Their example: The following hold.

$$\neg C \to B, (C \to B) \to D \vdash B \quad \text{and} \quad B, \neg C \to B, (C \to B) \to D \vdash D$$

but

$$\neg C \to B, (C \to B) \to D \nvdash D$$

A digression: naïvely adding NAF means deduction fails

First attempt: Following N-Prolog directly we could try the following naïve rules for fibering the systems of success and failure together using NAF.

$$\frac{\Delta \nvdash a}{\Delta \vdash \neg a}\,(\textit{to fail}) \qquad \frac{\Delta \vdash a}{\Delta \nvdash \neg a}\,(\textit{to succ})$$

An issue: it fails cut elimination (hence modus ponens)

> *"There are reasons to think that the addition of negation-as-failure to intuitionistic logic programming will lead to serious semantic diffculties. Gabbay studied the problem in [N-Prolog 1985], and concluded that the entire idea was logically incoherent, since modus ponens would no longer be valid." Boner and McCarty 1990*

Their example: The following hold.

$$\neg C \to B, (C \to B) \to D \vdash B \quad \text{and} \quad B, \neg C \to B, (C \to B) \to D \vdash D$$

but

$$\neg C \to B, (C \to B) \to D \nvdash D$$

The issue: The two instances of $\neg C$ refer to different databases.

Don't use these rules.

$$\frac{\Delta \nvdash a}{\Delta \vdash \neg a}\,(\mathit{to\ fail}) \qquad \frac{\Delta \vdash a}{\Delta \nvdash \neg a}\,(\mathit{to\ succ})$$

Use Gödel-Tarski constants to record the database used for by each occurence of negation

...duplicating clauses if used in different contexts (to be explained later).

The system fibering, presented next, becomes richer.

Second design step: add rules replacing $\top, \bot$ with Gödel-Tarski constants

Rules added to calculus of success:

$$\frac{}{\Delta, \bot \vdash A}(\bot) \quad \longrightarrow \quad \frac{\Gamma \nvdash B}{\Delta, \mathrm{Pr}(\Gamma, B) \vdash A}(\bot)$$

$$\frac{}{\Delta \vdash \top}(\top) \quad \longrightarrow \quad \frac{\Gamma \vdash B}{\Delta \vdash \mathrm{Pr}(\Gamma, B)}(\top)$$

$$\frac{\Delta \vdash A}{\Delta, A \to \bot \vdash q}(elim) \quad \longrightarrow \quad \frac{\Delta \vdash A \qquad \Gamma \nvdash B}{\Delta, A \to \mathrm{Pr}(\Gamma, B) \vdash q}(elim)$$

Remark: First and third rules fiber the calculus of failure into the calculus of success.

Second design step: add rules replacing $\top, \bot$ with Gödel-Tarski constants

Rules added to calculus of failure:

Old rule:

$$\frac{\left\{\Delta \nvdash A : \begin{array}{l}\text{for } A \text{ such that} \\ (A \Rightarrow x) \in \Delta, \text{ where } x = q \text{ or } x = \bot\end{array}\right\} \quad q \notin \Delta \quad \bot \notin \Delta}{\Delta \nvdash q}\;(\neg atom)$$

$$\downarrow$$

New Rule:

$$\frac{\begin{array}{cc}\left\{\Delta \nvdash A : \begin{array}{l}\text{for } A \text{ such that} \\ (A \Rightarrow x) \in \Delta \text{ where } x = q \text{ or } x = \bot\end{array}\right\} & q \notin \Delta \quad \bot \notin \Delta \\ \left\{\Delta \nvdash A \text{ or } \Gamma \vdash B : \begin{array}{l}\text{for } A \text{ such that} \\ (A \Rightarrow \mathrm{Pr}(\Gamma, B)) \in \Delta\end{array}\right\} & \{\Gamma \vdash B : \mathrm{Pr}(\Gamma, B) \in \Delta\}\end{array}}{\Delta \nvdash q}\;(\neg atom)$$

Remark: Fibers the calculus of success into the calculus of failure.

Note: The new rule replaces the old rule (in contrast to the system for success).

The rules with conclusions $\Delta \nvdash \bot$ or $\Delta \nvdash \mathrm{Pr}(\Gamma, B)$ are similar to the above.

Bonner & McCarty's example in this setting

We had:

$$\neg C \to B, (C \to B) \to D \vdash B \quad \text{and} \quad B, \neg C \to B, (C \to B) \to D \vdash D$$

Now we have for some X:

$$(\mathrm{Pr}(X, C) \to \bot) \to B, (C \to B) \to D \vdash B \quad \text{and} \quad B, (\mathrm{Pr}(X, C) \to \bot) \to B, (C \to B) \to D \vdash D$$

To preserve intuitionistic logic + NAF, we could set:

$$X = \quad (C \to B) \to D$$

...or more generally $X \nvdash C$.

Applying cut we obtain:

$$(\mathrm{Pr}(X, C) \to \bot) \to B, (C \to B) \to D \vdash D$$

Observation: This is now beyond N-Prolog, due to the jump to database X. We have moved to a richer logic to admit deduction.

Examples showing $\neg$ changes meaning.

Consider program consisting of (1) and (2):

1. $\neg c \to a$
2. $(c \to a) \to c$

Ask if a succeeds using NAF: $(1),(2) \vdash a$.

$$\begin{array}{c}
\hline
(1),(2),c \vdash c \\ \hline
(1),(2),c \nvdash \neg c \\ \hline
(1),(2),c \nvdash a \\ \hline
(1),(2) \nvdash c \to a \\ \hline
(1),(2) \nvdash c \\ \hline
(1),(2) \vdash \neg c \\ \hline
(1),(2) \vdash a
\end{array}$$

Examples showing $\neg$ changes meaning.

Consider program consisting of (1) and (2):

1. $\neg c \to a$
2. $(c \to a) \to c$

Ask if a succeeds using NAF: $(1),(2) \vdash a$.

$$\dfrac{\begin{array}{c} \hline (1),(2),c \vdash c \\ \hline (1),(2),c \nvdash \neg c \\ \hline (1),(2),c \nvdash a \\ \hline (1),(2) \nvdash c \to a \\ \hline (1),(2) \nvdash c \\ \hline (1),(2) \vdash \neg c \end{array}}{(1),(2) \vdash a}\ \text{clause (1)}$$

Examples showing ¬ changes meaning.

Consider program consisting of (1) and (2):

1. $\neg c \rightarrow a$
2. $(c \rightarrow a) \rightarrow c$

Ask if a succeeds using NAF: $(1),(2) \vdash a$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(1),(2),c \vdash c}}{(1),(2),c \nvdash \neg c}}{(1),(2),c \nvdash a}}{(1),(2) \nvdash c \rightarrow a}}{(1),(2) \nvdash c} \; \neg c = \mathrm{Pr}(\{(1),(2)\},c) \rightarrow \bot}{(1),(2) \vdash \neg c} \; \text{clause (1)}}{(1),(2) \vdash a}$$

Examples showing ¬ changes meaning.

Consider program consisting of (1) and (2):

1. $\neg c \to a$
2. $(c \to a) \to c$

Ask if a succeeds using NAF: $(1), (2) \vdash a$.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{(1),(2),c \vdash c}}{(1),(2),c \nvdash \neg c}\ \text{new instance of clause (1)}}{(1),(2),c \nvdash a}}{(1),(2) \nvdash c \to a}}{(1),(2) \nvdash c}\ \neg c = \mathrm{Pr}(\{(1),(2)\}, c) \to \bot}{(1),(2) \vdash \neg c}\ \text{clause (1)}}{(1),(2) \vdash a}
$$

Examples showing $\neg$ changes meaning.

Consider program consisting of (1) and (2):

1. $\neg c \to a$
2. $(c \to a) \to c$

Ask if a succeeds using NAF: $(1),(2) \vdash a$.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{(1),(2),c \vdash c}}{(1),(2),c \nvdash \neg c}\ \neg c = \mathrm{Pr}(\{(1),(2),c\},c) \to \bot}{(1),(2),c \nvdash a}\ \text{new instance of clause (1)}}{(1),(2) \nvdash c \to a}}{(1),(2) \nvdash c}}{(1),(2) \vdash \neg c}\ \neg c = \mathrm{Pr}(\{(1),(2)\},c) \to \bot}{(1),(2) \vdash a}\ \text{clause (1)}
$$

Examples showing ¬ changes meaning.

More explicit program replaces (1) with (1.1) and (1.2):

1. $\neg c \to a$
   - 1.1 $(\mathrm{Pr}(\{(1.2),(2)\},c) \to \bot) \to a$
   - 1.2 $(\mathrm{Pr}(\{c\},c) \to \bot) \to a$
2. $(c \to a) \to c$

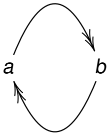
Ask if a succeeds using NAF: $(1.1),(1.2),(2) \vdash a$.

$$
\begin{array}{c}
\hline
c \vdash c \\
\hline
c, \mathrm{Pr}(\{c\},c) \nvdash \bot \\
\hline
c \nvdash \mathrm{Pr}(\{c\},c) \to \bot \\
\hline
(1.2), c \nvdash a \\
\hline
(1.2) \nvdash c \to a \\
\hline
(1.2),(2) \nvdash c \\
\hline
(1.2),(2),\mathrm{Pr}(\{(1.2),(2)\},c) \vdash \bot \\
\hline
(1.2),(2) \vdash \mathrm{Pr}(\{(1.2),(2)\},c) \to \bot \\
\hline
(1.1),(1.2),(2) \vdash a
\end{array}
$$

Remark: Consuming (2) avoids a cycle at the last step. In fact, to avoid cycles we can use as many copies of each clause as needed and then let our computation consume any clause it uses. So consuming resources acts as a loop checker, and adding resources acts as separating different occurences of ¬.

Example of resolving inconsistency

Consider a 2-cycle.



Model using the logic program with clauses:

- $(\mathrm{Pr}(X, a) \to \bot) \to b$
- $(\mathrm{Pr}(X, b) \to \bot) \to a$

Step 1: Pick a goal we want to hold, say a.

Step 2: Unfold the rules of our analytic proof calculus.

$$\dfrac{\dfrac{\dfrac{X \nvdash b}{(\mathrm{Pr}(X, a) \to \bot) \to b, (\mathrm{Pr}(X, b) \to \bot) \to a, \mathrm{Pr}(X, b) \vdash \bot}}{(\mathrm{Pr}(X, a) \to \bot) \to b, (\mathrm{Pr}(X, b) \to \bot) \to a \vdash \mathrm{Pr}(X, b) \to \bot}}{(\mathrm{Pr}(X, a) \to \bot) \to b, (\mathrm{Pr}(X, b) \to \bot) \to a \vdash a}$$

Remark: This is consistent with argumentation theory: if b is “out” then a is “in”.

An inconsistency that cannot be resolved.

Consider a self loop.

$$\begin{tikzcd} a \arrow[loop right] \end{tikzcd}$$

Model using the logic program with clauses:

- $(\mathrm{Pr}(X, a) \to \bot) \to a$

Step 1: Pick a goal we want to hold, say a.

Step 2: Unfold the rules of our analytic proof calculus.

$$\dfrac{\dfrac{\dfrac{X \nvdash a}{(\mathrm{Pr}(X, a) \to \bot) \to a, \mathrm{Pr}(X, a) \vdash \bot}}{(\mathrm{Pr}(X, a) \to \bot) \to a, \vdash \mathrm{Pr}(X, a) \to \bot}}{(\mathrm{Pr}(X, a) \to \bot) \to a \vdash a}$$

Remark: For goal a to be “in”, a must not be given resources. We loop if we try to make X consistent with the program, so cannot collapse between the goal and action levels.

Example of Inferring actions to take for a goal to succeed

Consider a chain.

$$a \twoheadrightarrow b \twoheadrightarrow c$$

Model using the logic program with clauses:

- $(\mathrm{Pr}(X, b) \to \bot) \to c$
- $(\mathrm{Pr}(X, a) \to \bot) \to b$

Step 1: Pick a goal we want to hold, say c.

Step 2: Unfold the rules of our analytic proof calculus.

$$\dfrac{\dfrac{\dfrac{X \nvdash b}{(\mathrm{Pr}(X, b) \to \bot) \to c, (\mathrm{Pr}(X, a) \to \bot) \to b, \mathrm{Pr}(X, b) \vdash \bot}}{(\mathrm{Pr}(X, b) \to \bot) \to c, (\mathrm{Pr}(X, a) \to \bot) \to b \vdash \mathrm{Pr}(X, b) \to \bot}}{(\mathrm{Pr}(X, b) \to \bot) \to c, (\mathrm{Pr}(X, a) \to \bot) \to b \vdash c}$$

Step 3: Force X to be consistent with our initial program.

$$\dfrac{\dfrac{\dfrac{X \vdash a}{(\mathrm{Pr}(X, b) \to \bot) \to c, (\mathrm{Pr}(X, a) \to \bot) \to b, \mathrm{Pr}(X, a) \nvdash \bot}}{(\mathrm{Pr}(X, b) \to \bot) \to c, (\mathrm{Pr}(X, a) \to \bot) \to b \nvdash \mathrm{Pr}(X, a) \to \bot}}{(\mathrm{Pr}(X, b) \to \bot) \to c, (\mathrm{Pr}(X, a) \to \bot) \to b \nvdash b}$$

Minimal solution: $X = a$ — give a resources and goal c will succeed.

Example, back to intuitionistic logic + ¬ with multiple contexts

Consider program:

- $((\neg a \to b) \to \neg b) \to c$
- $(b \to c) \to a$

Problem: Need two copies of first clause used to prove c fails **in different contexts**.

Find contexts X_1, Y_1, X_2, and Y_2 in the following program.

1. $(((\mathrm{Pr}(Y_1, a) \to \bot) \to b) \to (\mathrm{Pr}(X_1, b) \to \bot)) \to c$
2. $(((\mathrm{Pr}(Y_2, a) \to \bot) \to b) \to (\mathrm{Pr}(X_2, b) \to \bot)) \to c$
3. $(b \to c) \to a$

Taking $X_1 \vdash b$ we have a proof

$$\dfrac{\dfrac{\dfrac{\dfrac{X_1 \vdash b}{(2),(3),(\mathrm{Pr}(Y_1,a) \to \bot) \to b, \mathrm{Pr}(X_1,b) \vdash \to \bot}}{(2),(3),(\mathrm{Pr}(Y_1,a) \to \bot) \to b \vdash \mathrm{Pr}(X_1,b) \to \bot}}{(2),(3) \vdash ((\mathrm{Pr}(Y_1,a) \to \bot) \to b) \to (\mathrm{Pr}(X_1,b) \to \bot)}}{(1),(2),(3) \nvdash c}$$

Refining to $X_1 = (2),(3)\,(\mathrm{Pr}(Y_1,a) \to \bot) \to b$, we can restart the proof.

$$\dfrac{\dfrac{\dfrac{Y_1 \nvdash a}{(2),(3),\mathrm{Pr}(Y_1,a) \vdash \bot}}{(2),(3) \vdash \mathrm{Pr}(Y_1,a) \to \bot}}{(2),(3),(\mathrm{Pr}(Y_1,a) \to \bot) \to b \vdash b}$$

Which is a proof if $Y_1 \nvdash a$.

Example, back to intuitionistic logic + $\neg$ with multiple contexts

We have $X_1 = (2), (3), (\mathrm{Pr}(Y_1, a) \to \bot) \to b$, and $Y_1 \nvdash a$.
Let $Y_1 = (2), (3)$, so we can restart the proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{X_2 \vdash b}{b, (\mathrm{Pr}(Y_2, a) \to \bot) \to b, \mathrm{Pr}(X_2, b) \nvdash \bot}}{b, (\mathrm{Pr}(Y_2, a) \to \bot) \to b \nvdash \mathrm{Pr}(X_2, b) \to \bot}}{b \nvdash ((\mathrm{Pr}(Y_2, a) \to \bot) \to b) \to (\mathrm{Pr}(X_2, b) \to \bot)}}{(2), b \nvdash c}}{(2) \nvdash b \to c}}{(2), (3) \nvdash a}$$

Thus $X_2 \vdash b$.
Restarting with $X_2 = b, (\mathrm{Pr}(Y_2, a) \to \bot) \to b$ we conclude the proof.

$$\dfrac{}{b, (\mathrm{Pr}(Y_2, a) \to \bot) \to b \vdash b}$$

Thus there are no constraints on Y_2
Final solution, making proof analytic:

$$\begin{aligned} X_1 &= \quad (2), (3), (\mathrm{Pr}(Y_1, a) \to \bot) \to b \\ Y_1 &= \quad (2), (3) \\ X_2 &= \quad b, (\mathrm{Pr}(Y_2, a) \to \bot) \to b \\ Y_2 &= \quad \top \end{aligned}$$

proof theory

Analytic proof calculi are designed for algorithmic proof search and deduction (cut elimination).

KR

Negation as failure is useful for knowledge representation.

Common-sense problems can be addressed by making the context of different assertions explicit.

It is possible to have the best of both these worlds together in one system: a logic with $\neg$ in which the effect of cut is respected by cut elimination.

The system remembers from which database a query failed.

Future direction: this methodology will extend to any logic with well-founded proof system and constants for truth and falsity. Generating the rules for the system of failure should be achieved by systematically pushing $\neg$ to the atoms in deductions of the form $\Gamma \vdash \neg\Delta$.