

# A Propositional Plausible Logic

David Billington

School of Information and Communication Technology, Nathan campus,  
Griffith University, Brisbane, Queensland 4111, Australia.  
`d.billington@griffith.edu.au`

**Abstract.** A new non-monotonic propositional logic called PPL — for Propositional Plausible Logic — is defined. An example is worked and four theorems about PPL are stated.

**Keywords:** Plausible logic, Defeasible logic, Non-monotonic reasoning

## 1 Introduction

We shall define a new non-monotonic propositional logic called PPL, for Propositional Plausible Logic. PPL belongs to the family of propositional non-monotonic logics called Defeasible Logics, first defined in [2]. Unlike PPL, previous Defeasible Logics [1] do not handle conjunctions properly.

For a given formula, a proof algorithm will satisfy exactly one of the following four conditions.

- i) not terminate,
- ii) terminate in a state indicating that the formula is proved,
- iii) terminate in a state indicating that the formula is not provable,
- iv) terminate in some other state.

A proof algorithm is said to be **decisive** if and only if for every formula, the proof algorithm satisfies either (ii) or (iii). The proof algorithms of all previous Defeasible Logics were not decisive. All the proof algorithms of PPL are decisive, and so PPL is decidable.

This article is organised into the following sections. The definition of PPL is in Section 2. In Section 3 we apply PPL to an example. Four important theorems about PPL are stated and discussed in Section 4. Section 5 is the conclusion.

## 2 PPL: A Propositional Plausible Logic

PPL uses a propositional language — with negation  $\neg$ , conjunction  $\wedge$ , and disjunction  $\vee$  — to reason about facts and plausible information. The facts are represented by formulas that are converted into clauses called axioms, which are then converted into strict rules. The plausible information is represented by defeasible rules, warning rules, and a priority relation,  $>$ , on rules.

**Definition 1.** A rule,  $r$ , is any triple  $(A(r), \text{arrow}(r), c(r))$  such that  $A(r)$  is a finite set of formulas,  $\text{arrow}(r) \in \{\rightarrow, \Rightarrow, \rightsquigarrow\}$ , and  $c(r)$  is a formula. If  $r$  is a rule then  $A(r)$  is called the **set of antecedents** of  $r$ , and  $c(r)$  is called the **consequent** of  $r$ .

**Strict rules** use the **strict arrow**,  $\rightarrow$ , and are written  $A(r) \rightarrow c(r)$ .

**Defeasible rules** use the **defeasible arrow**,  $\Rightarrow$ , and are written  $A(r) \Rightarrow c(r)$ .

**Warning rules** use the **warning arrow**,  $\rightsquigarrow$ , and are written  $A(r) \rightsquigarrow c(r)$ .

Intuitively,  $A \rightarrow c$  means if every formula in  $A$  is true then  $c$  is true; whereas  $A \Rightarrow c$  means if every formula in  $A$  is true then  $c$  is usually true. Roughly,  $A \rightsquigarrow c$  means if every formula in  $A$  is true then  $c$  might be true. So  $A \rightsquigarrow \neg c$  warns against concluding usually  $c$ , but does not support usually  $\neg c$ .

**Definition 2.** Let  $R$  be any set of rules. A binary relation,  $>$ , on  $R$  is **cyclic** iff there exists a finite sequence,  $(r_1, r_2, \dots, r_n)$  where  $n \geq 1$ , of elements of  $R$  such that  $r_1 > r_2 > \dots > r_n > r_1$ .

A priority relation,  $>$ , on rules is used to indicate the more relevant of two rules. For instance the specific rule ‘Quails usually do not fly’ is more relevant than the general rule ‘Birds usually fly’ when reasoning about the flying ability of a quail. Hence ‘Quails usually do not fly’  $>$  ‘Birds usually fly’.

We shall now introduce some needed notation. The empty sequence is denoted by  $()$ . Let  $S$  be a sequence. If  $S$  is finite then  $S+e$  denotes the sequence formed by just adding  $e$  onto the right end of  $S$ . Define  $e \in S$  to mean  $e$  is an element of  $S$ , and  $e \notin S$  to mean  $e$  is not an element of  $S$ . A formula is **contingent** iff it is not a tautology and it is not a contradiction. If  $a$  is an atom then  $\sim a$  is  $\neg a$ , and  $\sim \neg a$  is  $a$ . If  $L$  is a set of literals then  $\sim L = \{\sim l : l \in L\}$ .

Next we define, among other things, the conversion of a clause to strict rules.

**Definition 3.** Let  $R$  be a set of rules,  $F$  be a finite set of formulas,  $C$  be a set of contingent clauses, and  $c$  be a contingent clause.

- 1)  $R_s$  is the set of strict rules in  $R$ .
- 2)  $R_d$  is the set of defeasible rules in  $R$ .
- 3)  $c(R)$  is the set of consequents of the rules in  $R$ .
- 4)  $Ru(c) = \{\{\} \rightarrow c\} \cup \{\{\wedge \sim(L-K)\} \rightarrow \forall K : c = \forall L \text{ and } \{\} \subset K \subset L\}$ .
- 5)  $Ru(C) = \bigcup \{Ru(c) : c \in C\}$ .
- 6)  $Ru(C, F)$  is the set of rules in  $Ru(C)$  whose set of antecedents is  $F$ .

If  $c$  has  $n$  literals then  $Ru(c)$  has  $2^n - 1$  strict rules. For example  $Ru(\forall\{a, b, c\}) = \{\{\} \rightarrow \forall\{a, b, c\}, \{\wedge\{\neg b, \neg c\}\} \rightarrow a, \{\wedge\{\neg a, \neg c\}\} \rightarrow b, \{\wedge\{\neg a, \neg b\}\} \rightarrow c, \{\neg a\} \rightarrow \forall\{b, c\}, \{\neg b\} \rightarrow \forall\{a, c\}, \{\neg c\} \rightarrow \forall\{a, b\}\}$ .

**Definition 4.** Let  $R$  be a finite set of rules.  $(R, >)$  is a **plausible theory** iff there is a satisfiable set of contingent clauses,  $Ax$ , such that 1, 2, and 3 hold.

- 1)  $R_s = \{A \rightarrow \wedge c(Ru(Ax, A)) : A \in \{A(r) : r \in Ru(Ax)\}\}$ .
- 2) If  $Ax \neq \{\}$  then  $r$  denotes the strict rule  $\{\} \rightarrow \wedge Ax$ .
- 3)  $>$  is a relation on  $R$  such that  $> \subseteq R \times (R - \{r\})$  and  $>$  is not cyclic.

Space limitations mean that we have made  $R$  finite, although this is not necessary. The rules in  $R_s$  are the ‘anding’ together of the rules in  $Ru(Ax)$  that have the same antecedent. The set  $Ax$  is the set of axioms of the plausible theory  $(R, >)$  and is denoted by  $Ax(R)$ .

In the rest of this section we define how to prove formulas given a plausible theory. This is done by naming the proof algorithms (Definition 5), defining the evidence for and against a formula (Definition 6), defining a mechanism for preventing proofs from looping (Definition 7), and finally defining the proof function and the proof algorithms (Definition 9).

**Definition 5.** Define  $Alg = \{\varphi, \pi, \psi, \beta, \beta', \psi', \pi'\}$  to be the **set of names of the proof algorithms**. Define  $\varphi' = \varphi$ . If  $\alpha \in \{\pi, \psi, \beta\}$  then define  $(\alpha')' = \alpha'' = \alpha$ . If  $\alpha \in Alg$ , then the **co-algorithm** of  $\alpha$  is  $\alpha'$ .

The  $\varphi$  algorithm mimics classical provability,  $\pi$  and  $\psi$  propagating ambiguity, and  $\beta$  blocks ambiguity, see Section 3. The co-algorithms are used to evaluate evidence against a formula.

The evidence for and against a formula is now defined.

**Definition 6.** Let  $(R, >)$  be a plausible theory,  $R' \subseteq R$ ,  $Ax = Ax(R)$ ,  $\alpha \in Alg$ ,  $f$  is a formula, and  $\{r, s\} \subseteq R$ .

- 1)  $R_d^s = (R_s \cup R_d) - \{\mathfrak{r}\}$ .
- 2)  $R'[f] = \{r \in R' : Ax \cup \{c(r)\} \text{ is satisfiable and } Ax \cup \{c(r)\} \models f\}$ .
- 3)  $R'[f; s] = \{t \in R'[f] : t > s\}$ .
- 4) If  $\alpha \in \{\varphi, \pi'\}$  or  $r = \mathfrak{r}$  then  $Foe(\alpha, f, r) = \{\}$ .  
 $Foe(\psi', f, r) = \{s \in R[\neg f] : s > r\} = R[\neg f; r]$ .  
If  $\alpha \notin \{\varphi, \pi', \psi'\}$  and  $r \neq \mathfrak{r}$  then  $Foe(\alpha, f, r) = \{s \in R[\neg f] : s \not\prec r\}$ .

Roughly  $R'[f]$  is the set of rules in  $R'$  whose consequents and axioms imply  $f$ . If  $f$  is not a fact then  $R_d^s[f]$  is all the evidence for  $f$ ; and  $Foe(\alpha, f, r)$  is the set of rules that  $\alpha$  regards as the evidence against  $f$  that is not inferior to  $r$ .

A history of used rules prevents proofs from looping.

**Definition 7.** Suppose  $(R, >)$  is a plausible theory and  $\alpha \in Alg$ . Define  $\alpha R = \{\alpha r : r \in R\}$ . Then  $H$  is an  $\alpha$ -**history** iff  $H$  is a finite sequence of elements of  $\alpha R \cup \alpha'R$  that has no repeated elements.

A formula is proved by evaluating its proof value, which is either  $+1$  or  $-1$ ;  $+1$  indicates the formula is proved, and  $-1$  indicates the formula cannot be proved. The arithmetic properties of the proof values are defined below. These are as expected, but note that  $\max\{-1, -1\} = -1$  and  $\min\{+1, +1\} = +1$ .

**Definition 8.** Suppose  $S \subseteq \{+1, -1\}$ . Then  $\min S = -1$  iff  $-1 \in S$ ,  
 $\min S = +1$  iff  $-1 \notin S$ ,  $\max S = +1$  iff  $+1 \in S$ ,  $\max S = -1$  iff  $+1 \notin S$ ,  
 $--1 = +1$ , and  $-+1 = -1$ .

The proof function  $P$  evaluates the proof value of its three arguments: the proof algorithm  $\alpha$ , the history of used rules  $H$ , and the formula  $f$  or set of formulas  $F$  to be evaluated. An explanation of each part of the following definition concludes this section.

**Definition 9.** Suppose  $\mathcal{P} = (R, >)$  is a plausible theory,  $Ax = Ax(R)$ ,  $\alpha \in Alg$ ,  $H$  is an  $\alpha$ -history, and  $f$  is a formula. The **proof function for  $\mathcal{P}$** ,  $P$ , and the proof algorithms are defined by P1 to P5.

P1) If  $F$  is a finite set of formulas, then  $P(\alpha, H, F) = \min\{P(\alpha, H, f) : f \in F\}$ .

P2) If  $Ax \models f$  then  $P(\alpha, H, f) = +1$ . Also  $P(\varphi, H, f) = +1$  iff  $Ax \models f$ .

P3) If  $Ax \not\models f$  and  $\alpha \neq \varphi$  then  

$$P(\alpha, H, f) = \max\{For(\alpha, H, f, r) : \alpha r \notin H \text{ and } r \in R_d^s[f]\}.$$

P4) If  $Ax \not\models f$ ,  $\alpha \neq \varphi$ ,  $\alpha r \notin H$ , and  $r \in R_d^s[f]$  then  $For(\alpha, H, f, r) = \min[\{P(\alpha, H + \alpha r, A(r))\} \cup \{Dftd(\alpha, H, f, r, s) : s \in Foe(\alpha, f, r)\}]$ .

P5) If  $Ax \not\models f$ ,  $\alpha \notin \{\varphi, \pi'\}$ ,  $\alpha r \notin H$ ,  $r \in R_d^s[f]$ , and  $s \in Foe(\alpha, f, r)$  then  

$$Dftd(\alpha, H, f, r, s) = \max[\{P(\alpha, H + \alpha t, A(t)) : \alpha t \notin H \text{ and } t \in R_d^s[f; s]\} \cup \{-P(\alpha', H + \alpha' s, A(s)) : \alpha' s \notin H\}].$$

If  $x$  is a formula or a finite set of formulas then define

$(\alpha, H) \vdash x$  iff  $P(\alpha, H, x) = +1$ , and  
 $\alpha \vdash x$  iff  $x$  is  $\alpha$ -provable iff  $P(\alpha, (), x) = +1$ .

To prove a set  $F$  of formulas every element of  $F$  must be proved; hence P1. If  $Ax \models f$  then  $f$  is a fact and so is declared proved. As  $\varphi$  only proves facts we have the second part of P2.

To prove  $f$  we need a rule for  $f$ ,  $r \in R_d^s[f]$ , that has not been used before,  $\alpha r \notin H$ , such that  $For(\alpha, H, f, r) = +1$ ; hence P3.  $For(\alpha, H, f, r) = +1$  whenever the set of antecedents of  $r$ ,  $A(r)$ , is proved; and when all the evidence against  $f$ , that is, rules in  $Foe(\alpha, f, r)$ , is defeated, that is,  $Dftd(\alpha, H, f, r, s) = +1$ . Whenever  $\alpha$  uses a rule  $r$ , we must add  $\alpha r$  to the history,  $H$ , of used rules giving  $H + \alpha r$ . Hence P4.

A rule  $s$  is defeated either by team defeat or by disabling  $s$ . The team of rules for  $f$  is  $R_d^s[f]$ . A rule  $s$  is defeated by team defeat if and only if there is a member  $t$  of the team for  $f$  such that  $t > s$ , that is,  $t \in R_d^s[f; s]$ , and every formula in the set of antecedents of  $t$ ,  $A(t)$ , is proved. Of course we must be sure that  $\alpha$  has not previously used  $t$ ,  $\alpha t \notin H$ , and we must add  $\alpha t$  to  $H$ . A rule  $s$  is disabled by showing that  $\alpha'$  cannot prove  $A(s)$ . Again we must be sure that  $\alpha'$  has not previously used  $s$ ,  $\alpha' s \notin H$ , and we must add  $\alpha' s$  to  $H$ . Hence P5.

Now that Plausible Logic is defined, it is time to apply it.

### 3 The Ambiguity Puzzle

Consider the following four statements.

- (1) There is evidence that  $a$  is usually true.
- (2) There is evidence that  $\neg a$  is usually true.
- (3) There is evidence that  $b$  is usually true.
- (4) If  $a$  is true then  $\neg b$  is usually true.

What can be concluded about  $b$ ? The evidence for  $b$  is (3). The evidence against  $b$  comes from (1) and (4). If we knew that  $a$  was definitely true then the evidence for  $b$  and against  $b$  would be equal. But (1) and (2) means that the evidence against  $b$  has been weakened.

Since (1) and (2) give equal evidence for and against  $a$ ,  $a$  is said to be **ambiguous**. If the evidence against  $b$  has been weakened sufficiently to allow  $b$  to be concluded, then  $b$  is not ambiguous. So the ambiguity of  $a$  has been blocked from propagating to  $b$ . An algorithm that can prove  $b$  (but not  $\neg b$ ) is said to be **ambiguity blocking**.

If the evidence against  $b$  has not been weakened sufficiently to allow  $b$  to be concluded, then  $b$  is ambiguous. So the ambiguity of  $a$  has been propagated to  $b$ . An algorithm that cannot prove  $b$  (or  $\neg b$ ) is said to be **ambiguity propagating**.

The plausible theory  $(R, >)$  which models the Ambiguity Puzzle is defined as follows. The priority relation  $>$  is empty, and  $R = \{r_a, r_{na}, r_b, r_{anb}\}$ , where  $r_a$  is  $\{\} \Rightarrow a$ ,  $r_{na}$  is  $\{\} \Rightarrow \neg a$ ,  $r_b$  is  $\{\} \Rightarrow b$ , and  $r_{anb}$  is  $\{a\} \Rightarrow \neg b$ .

Since  $R_s = \{\}$ ,  $Ax(R) = Ax = \{\}$ . So  $R[a] = \{r_a\}$ ,  $R[b] = \{r_b\}$ ,  $R[\neg a] = \{r_{na}\}$ , and  $R[\neg b] = \{r_{anb}\}$ . If  $l \in \{a, \neg a, b, \neg b\}$  and  $s \in R$  then  $R[l; s] = \{\}$ .

We shall show that  $\pi$  and  $\psi$  are ambiguity propagating and that  $\beta$  is ambiguity blocking. In the following evaluations we shall use  $\dagger$  and  $\square$ .

$\dagger$ )  $P(\alpha, H, \{f\}) = P(\alpha, H, f)$ , by P1.

$\square$ )  $P(\alpha, H, \{\}) = \min\{\} = +1$ , by P1.

**Evaluation E1.**  $\alpha \in \{\pi, \psi, \beta\}$

1 $\alpha$ )  $P(\alpha, (), b) = For(\alpha, (), b, r_b)$ , by P3

2 $\alpha$ )  $= \min\{P(\alpha, (\alpha r_b), \{\}), Dftd(\alpha, (), b, r_b, r_{anb})\}$ , by P4

3 $\alpha$ )  $= Dftd(\alpha, (), b, r_b, r_{anb})$ , by  $\square$

4 $\alpha$ )  $= -P(\alpha', (\alpha' r_{anb}), a)$ , by P5,  $\dagger$

5 $\alpha$ )  $= -For(\alpha', (\alpha' r_{anb}), a, r_a)$ , by P3

**Evaluation E2.**  $\alpha \in \{\pi, \psi\}$  and  $\alpha \not\vdash b$

5 $\alpha$ )  $P(\alpha, (), b) = -For(\alpha', (\alpha' r_{anb}), a, r_a)$ , by E1

6 $\alpha$ )  $= -P(\alpha', (\alpha' r_{anb}, \alpha' r_a), \{\})$ , by P4

7 $\alpha$ )  $= -1$ , by  $\square$ .

**Evaluation E3.**  $\beta \vdash b$

5 $\beta$ )  $P(\beta, (), b) = -For(\beta', (\beta' r_{anb}), a, r_a)$ , by E1

6 $\beta$ )  $= -\min\{P(\beta', (\beta' r_{anb}, \beta' r_a), \{\}), Dftd(\beta', (\beta' r_{anb}), a, r_a, r_{na})\}$ , by P4

7 $\beta$ )  $= -Dftd(\beta', (\beta' r_{anb}), a, r_a, r_{na})$ , by  $\square$

8 $\beta$ )  $= - - P(\beta, (\beta' r_{anb}, \beta r_{na}), \{\})$ , by P5

9 $\beta$ )  $= +1$ , by  $\square$ .

By Evaluation E2 and Theorem 1 (Decisiveness),  $\pi$  and  $\psi$  cannot prove  $b$  and so they are ambiguity propagating. By Evaluation E3 and Theorem 3 (2-Consistency),  $\beta$  proves  $b$  and so is ambiguity blocking.

## 4 Theorems about Propositional Plausible Logic

A major property of PPL is that it is decisive and hence decidable.

**Theorem 1 (Decisiveness).** Suppose  $\mathcal{P}$  is a plausible theory,  $\alpha \in Alg$ ,  $H$  is an  $\alpha$ -history, and  $x$  is either a formula or a finite set of formulas. Then either  $P(\alpha, H, x) = +1$  or  $P(\alpha, H, x) = -1$ , but not both.

Right Weakening is the property that if a formula  $f$  is provable and  $f$  classically implies a formula  $g$  then  $g$  is provable. Thus Right Weakening is closure under classical inference. The following result shows that both Right Weakening and ‘Modus Ponens (MP) for strict rules’ hold for PPL.

**Theorem 2 (Right Weakening).** Suppose  $(R, >)$  is a plausible theory,  $Ax = Ax(R)$ ,  $\alpha \in Alg$ ,  $H$  is an  $\alpha$ -history, and both  $f$  and  $g$  are formulas.

- 1) If  $(\alpha, H) \vdash f$  and  $Ax \cup \{f\} \models g$  then  $(\alpha, H) \vdash g$ .
- 2) If  $(\alpha, H) \vdash f$  and  $f \models g$  then  $(\alpha, H) \vdash g$ . (Right Weakening)
- 3) If  $A \rightarrow g \in R_s$  and  $(\alpha, H) \vdash A$  then  $(\alpha, H) \vdash g$ . (MP for strict rules)

The next theorem states that any two proved formulas are consistent with the axioms.

**Theorem 3 (2-Consistency).** Suppose  $(R, >)$  is a plausible theory,  $Ax = Ax(R)$ ,  $\alpha \in \{\varphi, \pi, \psi, \beta, \beta'\}$ , and both  $f$  and  $g$  are formulas. If  $\alpha \vdash f$  and  $\alpha \vdash g$  then  $Ax \cup \{f, g\}$  is satisfiable.

The final result shows that the proof algorithms form a hierarchy that is consistent with the intuition that ambiguity propagating proof algorithms are more cautious than ambiguity blocking algorithms.

**Theorem 4 (Hierarchy).** Suppose  $\mathcal{P} = (R, >)$  is a plausible theory. Let  $\mathcal{P}(\alpha)$  be the set of all formulas provable from  $\mathcal{P}$  using the proof algorithm  $\alpha$ .

- 1)  $\mathcal{P}(\varphi) \subseteq \mathcal{P}(\pi) \subseteq \mathcal{P}(\psi) \subseteq \mathcal{P}(\beta) = \mathcal{P}(\beta') \subseteq \mathcal{P}(\psi') \subseteq \mathcal{P}(\pi')$ .
- 2) If  $>$  is empty then  $\mathcal{P}(\varphi) \subseteq \mathcal{P}(\pi) = \mathcal{P}(\psi) \subseteq \mathcal{P}(\beta) = \mathcal{P}(\beta') \subseteq \mathcal{P}(\psi') = \mathcal{P}(\pi')$ .

So  $\beta'$ , the co-algorithm of  $\beta$ , proves exactly the same formulas as  $\beta$ . The set of formulas proved by  $\pi'$  is very similar to the union of all extensions of an extension based logic, like Default Logic or argumentation systems. The difference between  $\psi'$  and  $\pi'$  is that  $\psi'$  considers the priority relation  $>$  whereas  $\pi'$  does not. Since both  $\psi'$  and  $\pi'$  can prove a formula and its negation, it is better to think of them as evidence finders rather than algorithms that prove formulas are true.

## 5 Conclusion

A new propositional non-monotonic logic, called PPL for Propositional Plausible Logic, was defined. An example was worked and four theorems about PPL were stated. PPL has been implemented by George Wilson under the direction of Dr. Andrew Rock, who has implemented other Defeasible Logics.

Future work on PPL could involve a complexity analysis and a study of PPL’s implementation. It may be worthwhile relating PPL and argumentation systems. Adding variables to PPL would significantly increase its usefulness.

## Acknowledgement

The author thanks Patrick Marchisella for many helpful discussions.

## References

1. D. Billington (2011). *A Defeasible Logic for Clauses*. Lecture Notes in Artificial Intelligence vol. 7106 pages 472-480. Springer.
2. D. Nute (1987). *Defeasible Reasoning*. In *Proceedings of the 20th Hawaii International Conference on System Science* pages 470-477.