Changes to AProS Algorithm

Adam Conkey

July 22, 2010

$$\Gamma = \{x \mid x \text{ is a premise or assumption}\}$$

- Items colored green have been discussed and are ready to be implemented.
- Items colored red are possible suggestions in need of discussion.

1 Sentential Logic

1.1 Multiple Extractions

1.1.1 Extraction Path

When a goal has multiple extractions available, make choice by number of open questions. Follow extraction path until no more extractions exist, then count open questions.

- **Motiviation -** Suppose choice is made but new subgoal is extractable, extraction will be first strategy it attempts anyways. Making the choice of pursuing an extractable subgoal invariably leads to an extraction attempt.

- **Implementation -** Check to see if subgoal to be proved for an extraction is itself extractable. Continue until a non-extractable subgoal reached and then proceed with choice as usual.

- **Example**

1 of 67

1.	$(((P \& Q) \lor (R \& S)) \to D)$	Prem
2.	A	Prem
3.	$(A \to B)$	Prem
4.	$(B \to C)$	Prem
5.	$(C \to D)$	Prem
...	...	
6.	$((P \& Q) \lor (R \& S))$	***
7.	D	$\to$E 1, 6

There are two choices for extraction of goal D. In the first case $(P\&Q) \vee (R\&S)$ must be proved, which is in the end unprovable (AProS chooses this case as it sees $(P\&Q) \vee (R\&S)$ and C both as one open question and simply chooses the one occuring first in the proof). In the second case, one can trace the extractions as: D is extractable if C is provable, if this choice is made C is extractable if B is provable, B is extractable if A is provable, and A is trivially extractable. The second case reduces to proving A which has 0 open questions, compared to the other case which has 1 open question.

1.1.2 Choice by Complexity

If two extractions have the same number of open questions, make choice based on complexity of subgoals.

- **Motivation -** Though multiple extractions may have the same number of open questions, proving the subgoals may not be equivalent, i.e. one choice may generate more open questions when the subgoal is pursued than the other choice.
- **Implementation -** Rank formulae by complexity and choose the open question with the lower complexity. If there is not already a complexity ordering in AProS, this will require some thought.
- **Example**

1 of 65

1.	((P & (Q & (R & S))) → A)	Prem
2.	((T & U) → A)	Prem
3.	T	Prem
4.	U	Prem
...	...	
5.	(P & (Q & (R & S)))	***
6.	A	→E 1, 5

For the extraction of A, AProS ranks both choices as equivalent, i.e. $(P\&Q\&R\&S)$ and $(T\&U)$ are both considered to be one open question and it pursues the one occuring first in the proof. If ranked by complexity, it would have pursued the second case and proved it trivially. In general, ranking by complexity should at least reduce the number of open questions that will be generated in pursuing a choice, or generate easier subgoals to prove.

1.1.3 Choice by Syntactic Similarity

If choice of extraction is a tie by the above considerations, use syntactic similarity with assertions in Γ to differentiate further.

- **Motivation -** Syntactically informed decisions based on what is present in assumptions could produce distinguished choices that stand a better chance at being proved by what is already available in the proof.
- **Implementation -** Test syntactic similarity on assertions in Γ and base choice on the one that can produce the goal and has the greatest intersection of terms.
- **Example**

27 of 27

1.	$(\exists y)(\forall x)(x \in y \leftrightarrow x \subseteq a)$	Prem
2.	$(\forall x)(x \in u \leftrightarrow x \subseteq a)$	Assum
3.	$(\forall x)(x \in v \leftrightarrow x \subseteq a)$	Assum
4.	$z \in u$	Assum
5.	$(z \in v \leftrightarrow z \subseteq a)$	$\forall$E 3
6.	$(z \in u \leftrightarrow z \subseteq a)$	$\forall$E 2
7.	$z \subseteq a$	$\leftrightarrow$ER 6, 4
8.	$z \in v$	$\leftrightarrow$EL 5, 7
9.	$z \in v$	Assum
10.	$(z \in u \leftrightarrow z \subseteq a)$	$\forall$E 2
11.	$(z \in u \leftrightarrow z \subseteq a)$	$\forall$E 2
12.	$\neg z \in u$	Assum
13.	$(z \in u \leftrightarrow z \subseteq a)$	$\forall$E 2
14.	$(z \in v \leftrightarrow z \subseteq a)$	$\forall$E 3
15.	$z \subseteq a$	$\leftrightarrow$ER 14, 9
16.	$z \in u$	$\leftrightarrow$EL 13, 15
17.	$\perp$	$\perp$I 12, 16
18.	$z \in u$	$\neg$E 17
19.	$z \subseteq a$	$\leftrightarrow$ER 11, 18
20.	$z \in u$	$\leftrightarrow$EL 10, 19
21.	$(z \in u \leftrightarrow z \in v)$	$\leftrightarrow$I 8, 20
22.	$(\forall x)(x \in u \leftrightarrow x \in v)$	$\forall$I 21
23.	$u = v$	DefI (Extensionality) 22
24.	$((\forall x)(x \in v \leftrightarrow x \subseteq a) \rightarrow u = v)$	$\rightarrow$I 23
25.	$(\forall w)((\forall x)(x \in w \leftrightarrow x \subseteq a) \rightarrow u = w)$	$\forall$I 24
26.	$((\forall x)(x \in u \leftrightarrow x \subseteq a) \,\&\, (\forall w)((\forall x)(x \in w \leftrightarrow x \subseteq a) \rightarrow u = w))$	&I 2, 25
27.	$(\exists y)((\forall x)(x \in y \leftrightarrow x \subseteq a) \,\&\, (\forall w)((\forall x)(x \in w \leftrightarrow x \subseteq a) \rightarrow y = w))$	$\exists$I 26
28.	$(\exists y)((\forall x)(x \in y \leftrightarrow x \subseteq a) \,\&\, (\forall w)((\forall x)(x \in w \leftrightarrow x \subseteq a) \rightarrow y = w))$	$\exists$E 1, 27

There were multiple places from which $z \subseteq a$ (line 19)could have been extracted (lines 2,3, and 10). It obviously wouldn't make sense from 10, since that would just be looped reasoning, so it is essentially choosing between instantiating 2 or 3 appropriately and extracting $z \subseteq a$. AProS tried from line 2 and then went indirect (presumably because it already tried extraction previously on the same goal). It should have instead tried to instantiate and extract from line 3, and this would have been informed by the assumption in line 9 of $z \in v$ if the syntactic similarity were considered.

- **Note -** This idea might be somehow used in conjunction with the complexity strategy so that precedence isn't given to either, but rather the decision is made with trade-offs from both.

1.2 Division

When goal is atomic and there is a disjunction positively contained in Γ, division should only occur if the atomic goal is positively contained in Γ.

- **Motivation -** If the atomic goal is not positively contained in Γ, then it cannot be proved directly and the proof will at some point have to proceed indirectly. Doing so immediately prevents unnecessary search from multiple divisions and from proving the same indirect argument more than once.
- **Implementation**
  - Remove Division from available strategies if goal is atomic and is not positively contained in Γ; the next strategy to apply would then be Negation Elimination.
  - Can either test directly for positive occurrence or, if easier to implement and/or less computationally expensive, check for failed attempt at extraction, as an attempt at extraction implies a positive occurence of the goal.
- **Example**

1	$\neg(C \vee D)$	Premise
2	$D \leftrightarrow (E \vee F)$	Premise
3	$\neg A \rightarrow (C \vee F)$	Premise
4	$\vdots$	
5	A	Goal

A is not a positive subformula, but since there are disjunctions available, AProS attempts division. The resulting proof is 2831ss, 36L. It takes 2793 of those search steps to figure out the attempted divisions fail and then backtracks to try negation elimination, which then completes the proof quickly.

1.3 Conversion

1.3.1 Trivial Inversions

When goal is a disjunction and there are positively contained disjunctions in Γ, check trivial inversions on both disjuncts before proceeding to conversion.

- **Motivation -** If the goal disjunction can be trivially proven by an inversion on one of the disjuncts, it doesn't make sense to first make a case distinction with another disjunct only to use the trivial inversion anyways.
- **Implementation -** Before proceeding to conversion, check to see if one of the disjuncts is trivially extractable from Γ.
- **Example**

7 of 8

1.	(A ∨ B)	Prem
2.	(C ∨ D)	Prem
3.	A	Assum
4.	C	Assum
5.	(B ∨ A)	∨IL 3
6.	D	Assum
7.	(B ∨ A)	∨IL 3
8.	(B ∨ A)	∨E 2, 5, 7
9.	B	Assum
10.	C	Assum
11.	(B ∨ A)	∨IR 9
12.	D	Assum
13.	(B ∨ A)	∨IR 9
14.	(B ∨ A)	∨E 2, 11, 13
15.	(B ∨ A)	∨E 1, 8, 14

Obviously the disjunction $C \vee D$ is useless to the proof, but AProS utilizes it anyways.

1.3.2 Multiple Disjunctions

When there are multiple disjunctions in Γ for conversion, choose the disjunction that will result in the least number of open questions. If the goal disjunction has a common disjunct with a disjunction in Γ, the subderivation for the common disjunct should not contribute to the count of open questions.

- **Motivation -** If the goal disjunction has a disjunct in common with a positively contained disjunction in Γ, inversion in that case is trivial and will be detected by the trivial inversion check described in 1.2.1 above.
- **Implementation -** Use syntactic similarity between the goal disjunction and positively contained disjunctions in Γ to determine if a common disjunct exists. If yes, open question count is unchanged by subderivation involving the common disjunct.
- **Example**

1	$\neg(C \vee D)$	Premise
2	$D \leftrightarrow (E \vee F)$	Premise
3	$\neg A \rightarrow (C \vee F)$	Premise
4	$\quad \neg A$	Assume
5	$\quad \vdots$	
6	$\quad C \vee D$	Goal
7	$\quad \bot$	$\bot$I
8	A	$\neg$E

AProS should choose to use $C \vee F$ over $E \vee F$ for conversion on $C \vee D$ since they share C as a disjunct. Then by the trivial inversion check 1.2.1, it would immediately prove that subderivation and only have the extraction of the disjunction and the other subderivation left to prove.

2 Predicate Logic

2.1 Division

Same as suggestion in sentential case that was mentioned above, taking predicate letters to be atomic, e.g. if one has $J(u)$ as a goal, division should occur only if the predicate letter J has a positive occurrence in Γ.

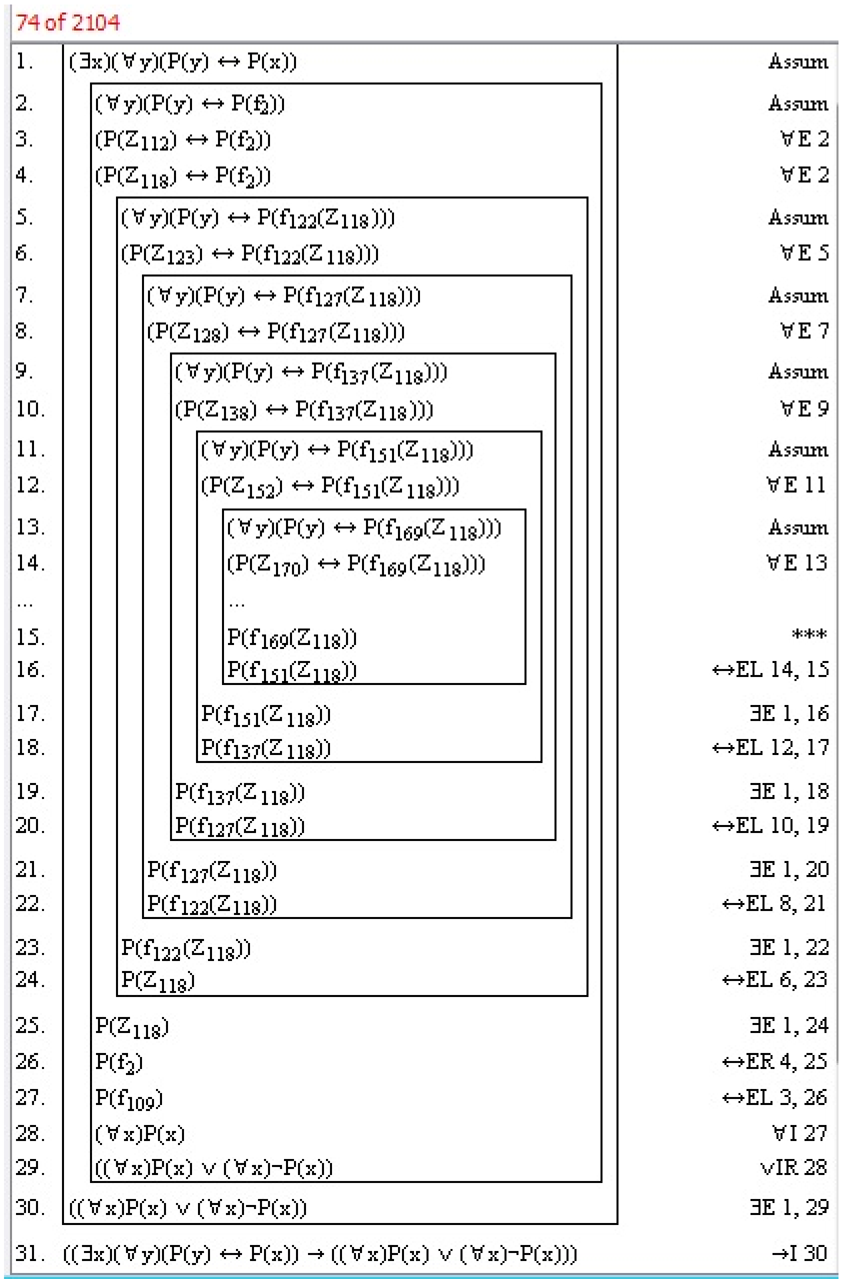
2.2 Existential Elimination

Existential Elimination should not occur more than once from any particular existentially quantified statement.

- **Motivation -** Since existentials only guarantee the existence of at least one object having the specified property, multiple eliminations from the same existential would either imply there are more than one distinct objects having that property (against what is intended by existential statements) or that the objects assumed to have the property are identical (introducing unnecessary equalities that complicate the search);

- **Implementation -** Once Existential Elimination is applied to an existential statement, remove it as a possibility for further existential eliminations. It should not become a possibility again unless the search backtracks beyond the point where the existential elimination occurred.

- **Example -** In most cases, if AProS does what is mentioned above, it backtracks immediately; it would be better though to bar the move from ever occurring as it should never pursue the proof further once it eliminates an existential more than once and necessarily has to backtrack a step. In extreme cases, AProS repeatedly does existential elimination, as in the following:



3 Predicate Logic with Equality

3.1 Self-Identity

All assertions of the form $x = x$ obtained from an application of $= I$ should be taken as trivial extractions. **(already implemented)**

- **Motivation -** It is a basic logical rule and shouldn't be considered an open question.
- **Implementation -** Consider all formulae of the form $x = x$ to be atomic.

3.2 Goal Equalities

For each goal equality, a choice should be presented between proving the equality itself or proving its symmetric counterpart.

- **Motivation -** Since equalities are inherently symmetric, it is arbitrary which way it is formulated as a goal, but the proof can be substantially more complicated pursuing one over the other.
- **Implementation -**
  - Treat goal equalities as disjunctions are treated for inversion, i.e. if the goal is $y = z$, it must choose whether to prove $y = z$ or $z = y$.
  - Choice should be determined as other choices are, by counting open questions.
  - If $z = y$ is chosen, apply one application of symmetry to $y = z$ and proceed to prove $z = y$.
  - If no advantage arises, i.e. open question count is the same, prove original goal of $y = z$.

4 Set Theory with Equality

4.1 Extensionality

Equality reasoning should be utilized over extensionality for goals expressing set-membership that are extractable from equalities.

- **Motivation -** Using extensionality adds unnecessary lines to the proof when equality reasoning can be trivially utilized.
- **Implementation -** When the goal is of this nature, place $=E$ higher in the strategy order than extractions from Extensionality.
- **Example** In the proof below, it should see for the goal of y=z that it has an extraction from the definition of singleton as it does since $\{z\}$ occurs in an assumption, but instead of pursuing that extraction through extensionality it should apply $=E$ on line 5 using line 1 to obtain line 4 as a goal without ever appealing to extensionality. The proof would then be the same with lines 2 and 3 omitted.

5 of 44

1.	$\{x,y\} = \{z\}$	Assum
2.	$(\forall z_2)(z_2 \in \{x,y\} \leftrightarrow z_2 \in \{z\})$	DefE (Extensionality) 1
3.	$(Z_{92} \in \{x,y\} \leftrightarrow Z_{92} \in \{z\})$	$\forall$E 2
...	...	
4.	$Z_{92} \in \{x,y\}$	***
5.	$Z_{92} \in \{z\}$	$\leftrightarrow$ER 3, 4
6.	$y = z$	DefE (Singleton) 5
7.	$(\{x,y\} = \{z\} \rightarrow y = z)$	$\rightarrow$I 6