Composable Discovery Engines for Interactive Theorem Proving

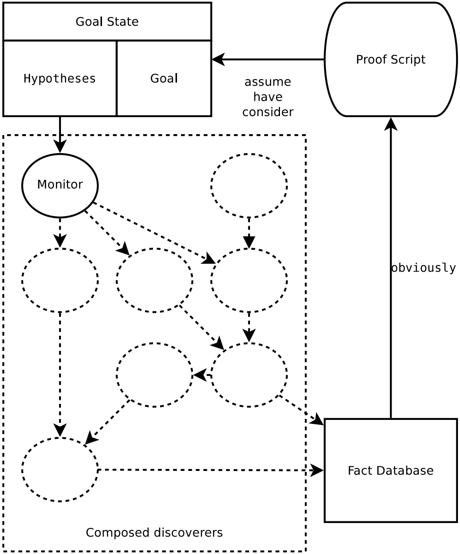
Phil Scott and Jacques Fleuriot

August 22, 2011

Overview

- Framework
- Search Algebra
- Proof trees
- Integration with Theorem Proving
- Issues
- Example

Collaborative Discovery



Search Algebra[1]

Given

$$\lambda x.\{x\} : \alpha \rightarrow \alpha\ bag$$
$$map : (\alpha \rightarrow \beta) \rightarrow \alpha\ bag \rightarrow \beta\ bag$$
$$\emptyset : bag$$
$$\cup : bag \rightarrow bag \rightarrow bag$$

Define

$$\alpha\ stream := \alpha\ bag\ list$$
$$unit\ x := [\{x\}, \emptyset, \emptyset \ldots]$$
$$map\_stream\ f\ [x_1, x_2, x_3] := [map\ f\ x_1, map\ f\ x_2, map\ f\ x_3, \ldots]$$
$$0 := [\emptyset, \emptyset, \emptyset, \ldots]$$
$$[x_1, x_2, x_3, \ldots] + [y_1, y_2, y_3, \ldots] := [x_1 \cup y_1, x_2 \cup y_2, x_3 \cup y_3, \ldots]$$
$$delay\ [x_1, x_2, x_3, \ldots] := [\emptyset, x_1, x_2, x_3, \ldots]$$
$$fix\ f\ xs := ys \quad where \quad ys := xs \cup delay\ (f\ ys)$$

[1] See Spivey's *Algebras for Combinatorial Search*

Search Algebras Continued

Want $join : \alpha\ stream\ stream \rightarrow \alpha\ stream$

For now $\alpha\ bag\ list\ list \rightarrow \alpha\ bag\ list$

$$
\begin{array}{lccccccccc}
[[& a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, & \ldots \\
[& b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, & b_7, & \ldots \\
[& c_0, & c_1, & c_2, & c_3, & c_4, & c_5, & c_6, & c_7, & \ldots \\
[& d_0, & d_1, & d_2, & d_3, & d_4, & d_5, & d_6, & d_7, & \ldots \\
[& e_0, & e_1, & e_2, & e_3, & e_4, & e_5, & e_6, & e_7, & \ldots \\
[& f_0, & f_1, & f_2, & f_3, & f_4, & f_5, & f_6, & f_7, & \ldots \\
[& g_0, & g_1, & g_2, & g_3, & g_4, & g_5, & g_6, & g_7, & \ldots \\
[& h_0, & h_1, & h_2, & h_3, & h_4, & h_5, & h_6, & h_7, & \ldots \\
 & & \vdots & & & & & & &
\end{array}
$$

Search Algebras Continued

Want $join : \alpha\ stream\ stream \rightarrow \alpha\ stream$

For now $\alpha\ bag\ list\ list \rightarrow \alpha\ bag\ list$

$$
\begin{array}{lllllllll}
[[& a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, \ldots \\
[& & b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, \ldots \\
[& & c_0, & c_1, & c_2, & c_3, & c_4, & c_5, & c_6, \ldots \\
[& & d_0, & d_1, & d_2, & d_3, & d_4, & d_5, & d_6, \ldots \\
[& & e_0, & e_1, & e_2, & e_3, & e_4, & e_5, & e_6, \ldots \\
[& & f_0, & f_1, & f_2, & f_3, & f_4, & f_5, & f_6, \ldots \\
[& & g_0, & g_1, & g_2, & g_3, & g_4, & g_5, & g_6, \ldots \\
[& & h_0, & h_1, & h_2, & h_3, & h_4, & h_5, & h_6, \ldots \\
 & & & \vdots & & & & &
\end{array}
$$

Search Algebras Continued

Want $join : \alpha\ stream\ stream \rightarrow \alpha\ stream$

For now $\alpha\ bag\ list\ list \rightarrow \alpha\ bag\ list$

$$
\begin{array}{lllllllll}
[[& a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, \ \dots \\
 & [& b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, \ \dots \\
 & & [& c_0, & c_1, & c_2, & c_3, & c_4, & c_5, \ \dots \\
 & & [& d_0, & d_1, & d_2, & d_3, & d_4, & d_5, \ \dots \\
 & & [& e_0, & e_1, & e_2, & e_3, & e_4, & e_5, \ \dots \\
 & & [& f_0, & f_1, & f_2, & f_3, & f_4, & f_5, \ \dots \\
 & & [& g_0, & g_1, & g_2, & g_3, & g_4, & g_5, \ \dots \\
 & & [& h_0, & h_1, & h_2, & h_3, & h_4, & h_5, \ \dots \\
 & & & & \vdots & & & &
\end{array}
$$

Search Algebras Continued

Want $join : \alpha\ stream\ stream \rightarrow \alpha\ stream$

For now $\alpha\ bag\ list\ list \rightarrow \alpha\ bag\ list$

$$
\begin{array}{llllllllll}
[[& a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, & \dots \\
 & [& b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, & \dots \\
 & & [& c_0, & c_1, & c_2, & c_3, & c_4, & c_5, & \dots \\
 & & & [& d_0, & d_1, & d_2, & d_3, & d_4, & \dots \\
 & & & [& e_0, & e_1, & e_2, & e_3, & e_4, & \dots \\
 & & & [& f_0, & f_1, & f_2, & f_3, & f_4, & \dots \\
 & & & [& g_0, & g_1, & g_2, & g_3, & g_4, & \dots \\
 & & & [& h_0, & h_1, & h_2, & h_3, & h_4, & \dots \\
 & & & & & \vdots & & & &
\end{array}
$$

Search Algebras Continued

Want $join : \alpha\ stream\ stream \rightarrow \alpha\ stream$

For now $\alpha\ bag\ list\ list \rightarrow \alpha\ bag\ list$

$$
\begin{array}{lllllllll}
[[& a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, \ldots \\
& [& b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, \ldots \\
& & [& c_0, & c_1, & c_2, & c_3, & c_4, & c_5, \ldots \\
& & & [& d_0, & d_1, & d_2, & d_3, & d_4, \ldots \\
& & & & [& e_0, & e_1, & e_2, & e_3, \ldots \\
& & & & [& f_0, & f_1, & f_2, & f_3, \ldots \\
& & & & [& g_0, & g_1, & g_2, & g_3, \ldots \\
& & & & [& h_0, & h_1, & h_2, & h_3, \ldots \\
& & & & & & \vdots & &
\end{array}
$$

Search Algebras Continued

$$\text{Want} \quad join : \alpha\ stream\ stream \rightarrow \alpha\ stream$$

$$\text{For now} \quad \alpha\ bag\ list\ list \rightarrow \alpha\ bag\ list$$

$$\begin{array}{llllllllll}
[[& a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, & \ldots \\
 & [& b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, & \ldots \\
 & & [& c_0, & c_1, & c_2, & c_3, & c_4, & c_5, & \ldots \\
 & & & [& d_0, & d_1, & d_2, & d_3, & d_4, & \ldots \\
 & & & & [& e_0, & e_1, & e_2, & e_3, & \ldots \\
 & & & & & [& f_0, & f_1, & f_2, & \ldots \\
 & & & & & [& g_0, & g_1, & g_2, & \ldots \\
 & & & & & [& h_0, & h_1, & h_2, & \ldots \\
 & & & & & & & \vdots & &
\end{array}$$

Search Algebras Continued

Want $\;join : \alpha\; stream\; stream \rightarrow \alpha\; stream$

For now $\;\alpha\; bag\; list\; list \rightarrow \alpha\; bag\; list$

$$
\begin{array}{llllllllll}
[[& a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, & \dots \\
 & [& b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, & \dots \\
 & & [& c_0, & c_1, & c_2, & c_3, & c_4, & c_5, & \dots \\
 & & & [& d_0, & d_1, & d_2, & d_3, & d_4, & \dots \\
 & & & & [& e_0, & e_1, & e_2, & e_3, & \dots \\
 & & & & & [& f_0, & f_1, & f_2, & \dots \\
 & & & & & & [& g_0, & g_1, & \dots \\
 & & & & & & [& h_0, & h_1, & \dots \\
 & & & & & & & & \vdots &
\end{array}
$$

Search Algebras Continued

Want $join : \alpha\ stream\ stream \rightarrow \alpha\ stream$

For now $\alpha\ bag\ list\ list \rightarrow \alpha\ bag\ list$

$$
\begin{array}{cccccccccc}
[[& a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, & \ldots \\
 & [& b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, & \ldots \\
 & & [& c_0, & c_1, & c_2, & c_3, & c_4, & c_5, & \ldots \\
 & & & [& d_0, & d_1, & d_2, & d_3, & d_4, & \ldots \\
 & & & & [& e_0, & e_1, & e_2, & e_3, & \ldots \\
 & & & & & [& f_0, & f_1, & f_2, & \ldots \\
 & & & & & & [& g_0, & g_1, & \ldots \\
 & & & & & & & [& h_0, & \ldots \\
 & & & & & & & & & \vdots
\end{array}
$$

Search Algebras Continued

Want $join : \alpha\ stream\ stream \to \alpha\ stream$

For now $\alpha\ bag\ list\ list \to \alpha\ bag\ list$

[[	a_0,	a_1,	a_2,	a_3,	a_4,	a_5,	a_6,	a_7,	...
	[	b_0,	b_1,	b_2,	b_3,	b_4,	b_5,	b_6,	...
		[	c_0,	c_1,	c_2,	c_3,	c_4,	c_5,	...
			[	d_0,	d_1,	d_2,	d_3,	d_4,	...
				[	e_0,	e_1,	e_2,	e_3,	...
					[	f_0,	f_1,	f_2,	...
						[	g_0,	g_1,	...
							[	h_0,	...
									$\vdots$

Search Algebras Continued

Want $join : \alpha\ stream\ stream \to \alpha\ stream$

For now $\alpha\ bag\ list\ list \to \alpha\ bag\ list$

$$
\begin{array}{cccccccccc}
[[& a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, & \ldots \\
 & [& b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, & \ldots \\
 & & [& c_0, & c_1, & c_2, & c_3, & c_4, & c_5, & \ldots \\
 & & & [& d_0, & d_1, & d_2, & d_3, & d_4, & \ldots \\
 & & & & [& e_0, & e_1, & e_2, & e_3, & \ldots \\
 & & & & & [& f_0, & f_1, & f_2, & \ldots \\
 & & & & & & [& g_0, & g_1, & \ldots \\
 & & & & & & & [& h_0, & \ldots \\
 & & & & & & & & & \vdots
\end{array}
$$

Search Algebras Continued

Want $join : \alpha\ stream\ stream \rightarrow \alpha\ stream$

For now $\alpha\ bag\ list\ list \rightarrow \alpha\ bag\ list$

$$
\begin{array}{cccccccccc}
[[& a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, & \ldots \\
 & [& b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, & \ldots \\
 & & [& c_0, & c_1, & c_2, & c_3, & c_4, & c_5, & \ldots \\
 & & & [& d_0, & d_1, & d_2, & d_3, & d_4, & \ldots \\
 & & & & [& e_0, & e_1, & e_2, & e_3, & \ldots \\
 & & & & & [& f_0, & f_1, & f_2, & \ldots \\
 & & & & & & [& g_0, & g_1, & \ldots \\
 & & & & & & & [& h_0, & \ldots \\
 & & & & & & & & & \vdots
\end{array}
$$

Search Algebras Continued

Want $join : \alpha\ stream\ stream \rightarrow \alpha\ stream$

For now $\alpha\ bag\ list\ list \rightarrow \alpha\ bag\ list$

$$
\begin{array}{cccccccccc}
[[& a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, & \ldots \\
 & [& b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, & \ldots \\
 & & [& c_0, & c_1, & c_2, & c_3, & c_4, & c_5, & \ldots \\
 & & & [& d_0, & d_1, & d_2, & d_3, & d_4, & \ldots \\
 & & & & [& e_0, & e_1, & e_2, & e_3, & \ldots \\
 & & & & & [& f_0, & f_1, & f_2, & \ldots \\
 & & & & & & [& g_0, & g_1, & \ldots \\
 & & & & & & & [& h_0, & \ldots \\
 & & & & & & & & & \vdots
\end{array}
$$

Search Algebras Continued

Want $\mathit{join} : \alpha\ \mathit{stream}\ \mathit{stream} \rightarrow \alpha\ \mathit{stream}$

For now $\alpha\ \mathit{bag}\ \mathit{list}\ \mathit{list} \rightarrow \alpha\ \mathit{bag}\ \mathit{list}$

$$
\begin{array}{cccccccccc}
[[& a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, & \dots \\
 & [& b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, & \dots \\
 & & [& c_0, & c_1, & c_2, & c_3, & c_4, & c_5, & \dots \\
 & & & [& d_0, & d_1, & d_2, & d_3, & d_4, & \dots \\
 & & & & [& e_0, & e_1, & e_2, & e_3, & \dots \\
 & & & & & [& f_0, & f_1, & f_2, & \dots \\
 & & & & & & [& g_0, & g_1, & \dots \\
 & & & & & & & [& h_0, & \dots \\
 & & & & & & & & & \vdots
\end{array}
$$

Search Algebras Continued

Want *join* : α *stream stream* $\rightarrow$ α *stream*

For now α *bag list list* $\rightarrow$ α *bag list*

$$
\begin{array}{llllllllll}
[[& a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, & \ldots \\
 & [& b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, & \ldots \\
 & & [& c_0, & c_1, & c_2, & c_3, & c_4, & c_5, & \ldots \\
 & & & [& d_0, & d_1, & d_2, & d_3, & d_4, & \ldots \\
 & & & & [& e_0, & e_1, & e_2, & e_3, & \ldots \\
 & & & & & [& f_0, & f_1, & f_2, & \ldots \\
 & & & & & & [& g_0, & g_1, & \ldots \\
 & & & & & & & [& h_0, & \ldots \\
 & & & & & & & & & \vdots
\end{array}
$$

Search Algebras Continued

Want $join : \alpha\ stream\ stream \rightarrow \alpha\ stream$

For now $\alpha\ bag\ list\ list \rightarrow \alpha\ bag\ list$

$$
\begin{array}{cccccccccc}
[[& a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, & \ldots \\
 & [& b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, & \ldots \\
 & & [& c_0, & c_1, & c_2, & c_3, & c_4, & c_5, & \ldots \\
 & & & [& d_0, & d_1, & d_2, & d_3, & d_4, & \ldots \\
 & & & & [& e_0, & e_1, & e_2, & e_3, & \ldots \\
 & & & & & [& f_0, & f_1, & f_2, & \ldots \\
 & & & & & & [& g_0, & g_1, & \ldots \\
 & & & & & & & [& h_0, & \ldots \\
 & & & & & & & & & \vdots
\end{array}
$$

Search Algebras Continued

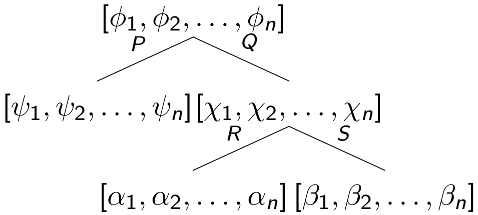
Want $join : \alpha\ stream\ stream \rightarrow \alpha\ stream$

For now $\alpha\ bag\ list\ list \rightarrow \alpha\ bag\ list$

$$
\begin{array}{cccccccccc}
[[& a_0, & a_1, & a_2, & a_3, & a_4, & a_5, & a_6, & a_7, & \ldots \\
 & [& b_0, & b_1, & b_2, & b_3, & b_4, & b_5, & b_6, & \ldots \\
 & & [& c_0, & c_1, & c_2, & c_3, & c_4, & c_5, & \ldots \\
 & & & [& d_0, & d_1, & d_2, & d_3, & d_4, & \ldots \\
 & & & & [& e_0, & e_1, & e_2, & e_3, & \ldots \\
 & & & & & [& f_0, & f_1, & f_2, & \ldots \\
 & & & & & & [& g_0, & g_1, & \ldots \\
 & & & & & & & [& h_0, & \ldots \\
 & & & & & & & & & \vdots
\end{array}
$$

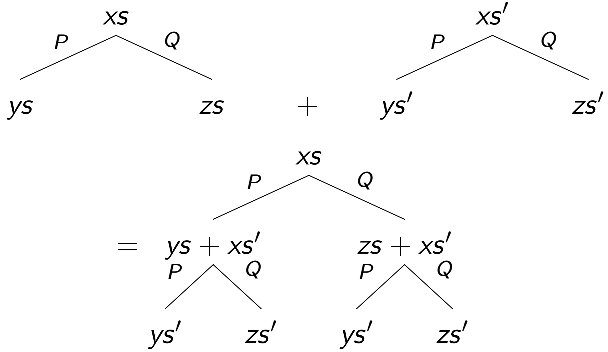
Case-splitting

$$\phi_1 \wedge \phi_2 \wedge \cdots \wedge \phi_n \wedge (P \rightarrow \psi_1 \wedge \psi_2 \wedge \cdots \wedge \psi_n) \wedge (Q \rightarrow \chi_1 \wedge \chi_2 \wedge \cdots \wedge \chi_n$$
$$\wedge\, (R \rightarrow \alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n) \wedge (S \rightarrow \beta_1 \wedge \beta_2 \wedge \cdots \wedge \beta_n))$$

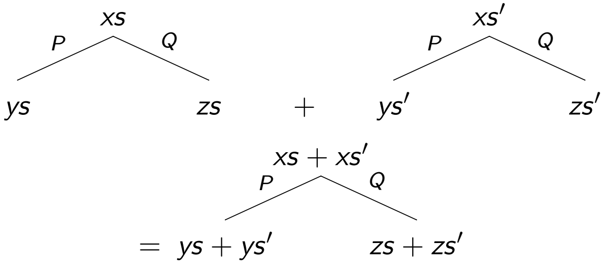


Replacing bags with tagged trees gives a new kind of search type:
α *chain* $:=$ α *tree list*

Combining Case-Splits



Combining Case-Splits



Integration

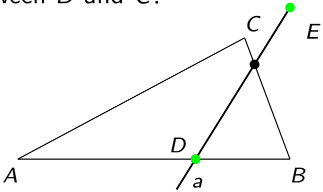
`monitor`	*thm chain*
`gen`	*term* $\longrightarrow$ *thm chain* $\longrightarrow$ *thm chain*
`consider`	*thm chain* $\longrightarrow$ *thm chain*
`conjuncts`	*thm chain* $\longrightarrow$ *thm chain*
`rewrite`	*thm list* $\longrightarrow$ *thm chain* $\longrightarrow$ *thm chain*
`mp`	*thm chain* $\longrightarrow$ *thm chain* $\longrightarrow$ *thm chain*
`chain1`	*thm* $\longrightarrow$ *thm chain* $\longrightarrow$ *thm chain*
`chain2`	*thm* $\longrightarrow$ *thm chain* $\longrightarrow$ *thm chain* $\longrightarrow$ *thm chain*
`chain3`	*thm* $\longrightarrow$ *thm chain* $\longrightarrow$ *thm chain* $\longrightarrow$ *thm chain* $\longrightarrow$ *thm chain*

Ocaml Issues

- Weak multi-threading
- Data-structures such as lazy-lists are not thread-safe
- Interrupts!
- Laziness subtleties

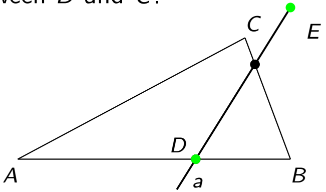
Pasch's Axiom

If a line a intersects a triangle ABC between A and B, and does not intersect any of A, B or C, then it intersects the triangle either between A and C or between B and C.



Pasch's Axiom

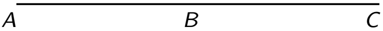
If a line a intersects a triangle ABC between A and B, and does not intersect any of A, B or C, then it intersects the triangle either between A and C or between B and C.



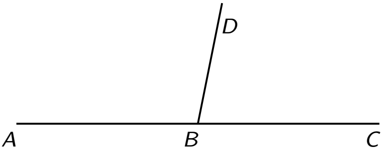
Conditions:

- ABC is a triangle
- ADE is a triangle
- BDE is a triangle
- CDE is a triangle

Theorem 4

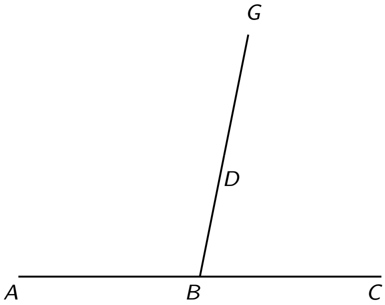


Theorem 4



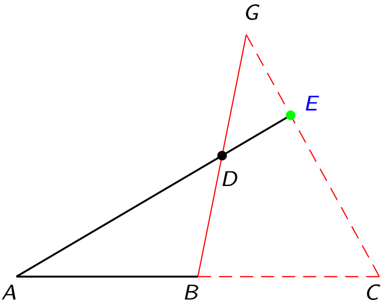
- Find D off the line AC.

Theorem 4



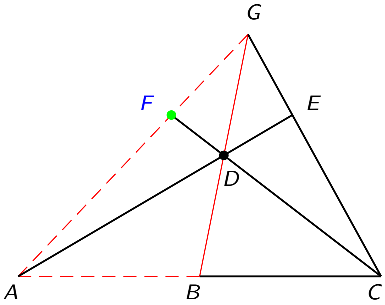
- Find D off the line AC.
- Extend BD to G

Theorem 4



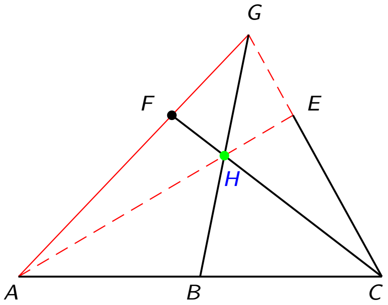
- Find D off the line AC.
- Extend BD to G
- Use Pasch on AD and BCG to obtain E between C and G

Theorem 4



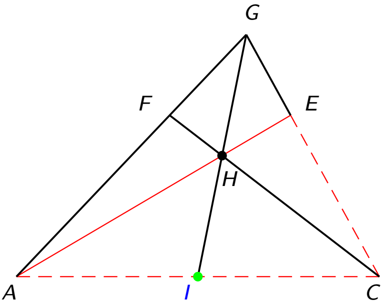
- Find D off the line AC.
- Extend BD to G
- Use Pasch on AD and BCG to obtain E between C and G
- Use Pasch on CD and ABG to obtain F between A and G

Theorem 4



- Find D off the line AC.
- Extend BD to G
- Use Pasch on AD and BCG to obtain E between C and G
- Use Pasch on CD and ABG to obtain F between A and G
- Use Pasch on CF and AEG to obtain H between A and E

Theorem 4



- Find D off the line AC.
- Extend BD to G
- Use Pasch on AD and BCG to obtain E between C and G
- Use Pasch on CD and ABG to obtain F between A and G
- Use Pasch on CF and AEG to obtain H between A and E
- Use Pasch on BG and ACE to obtain I between A and C.

Theorem 4

Formalised Proof

```
prove collinear {A, B, C} ∧ A ≠ B ∧ A ≠ C ∧ B ≠ C
        ∧ ¬ between A C B ∧ ¬between B A C ⟹ between A B C
assume collinear {A, B, C} ∧ A ≠ B ∧ A ≠ C ∧ B ≠ C
        ∧ ¬ between A C B ∧ ¬between B A C
so consider D such that ¬collinear {A, B, D}
        by construct_triangle
3 triangles discovered
obviously consider G such that between B D G by g22
8 triangles discovered
consider E such that collinear {A, D, E} ∧ between C E G
        by pasch_on B,C,G and A,D
16 triangles discovered
consider F such that collinear {C, D, F} ∧ between A F G
        by pasch_on A,B,G and C,D
28 triangles discovered
have between A D E by pasch_on A,E,G and C,F
have between A B C by pasch_on A,C,E and B,G
qed
```

Conclusion and Further Work

- Automated collaborative discovery.
- Discovery exploits idle time for complex combinatorial reasoning.
- Search-algebra allows advanced users to write their own discovery engines.

Future work

- Filtering/subsumption
- Proof replay without idle-time