A Curiously Effective Backtracking Strategy for Connection Tableaux

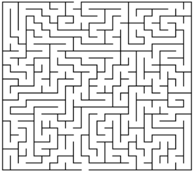
Michael Färber

September 18, 2023

Section 1

Introduction

Introduction



- Proof search with the connection calculus frequently uses backtracking
- Restricted backtracking is incomplete, but frequently increases the number of proofs found in given time

This Talk

- *Less restricted backtracking* is a new backtracking strategy
- It lies between restricted and unrestricted backtracking
- It is implemented in the new connection prover meanCoP
- On most evaluated datasets, it proves more problems than any other strategy

Section 2

Example

Proof Search (Unrestricted Backtracking)

$$\begin{bmatrix} \begin{bmatrix} \mathsf{p}(x) \\ \mathsf{q}(x) \end{bmatrix} & \begin{bmatrix} \neg\mathsf{p}(y) \\ \mathsf{r}(y) \end{bmatrix} & \begin{bmatrix} \neg\mathsf{p}(z) \end{bmatrix} & \begin{bmatrix} \neg\mathsf{r}(\mathsf{a}) \end{bmatrix} & \begin{bmatrix} \neg\mathsf{r}(\mathsf{b}) \end{bmatrix} & \begin{bmatrix} \neg\mathsf{q}(\mathsf{c}) \end{bmatrix} \end{bmatrix} \tag{1}$$

Proof Search (Unrestricted Backtracking)

$$\left[\begin{bmatrix} \mathrm{p}(x) \\ \mathrm{q}(x) \end{bmatrix} \begin{bmatrix} \neg\mathrm{p}(y) \\ \mathrm{r}(y) \end{bmatrix} \begin{bmatrix} \neg\mathrm{p}(z) \end{bmatrix} \begin{bmatrix} \neg\mathrm{r}(\mathrm{a}) \end{bmatrix} \begin{bmatrix} \neg\mathrm{r}(\mathrm{b}) \end{bmatrix} \begin{bmatrix} \neg\mathrm{q}(\mathrm{c}) \end{bmatrix}\right] \tag{1}$$

Proof Search (Unrestricted Backtracking)

$$\left[\begin{bmatrix} \mathrm{p}(x) \\ \mathrm{q}(x) \end{bmatrix} \begin{bmatrix} \neg\mathrm{p}(y) \\ \mathrm{r}(y) \end{bmatrix} \begin{bmatrix} \neg\mathrm{p}(z) \end{bmatrix} \begin{bmatrix} \neg\mathrm{r}(\mathrm{a}) \end{bmatrix} \begin{bmatrix} \neg\mathrm{r}(\mathrm{b}) \end{bmatrix} \begin{bmatrix} \neg\mathrm{q}(\mathrm{c}) \end{bmatrix} \right] \tag{1}$$

Proof Search (Unrestricted Backtracking)

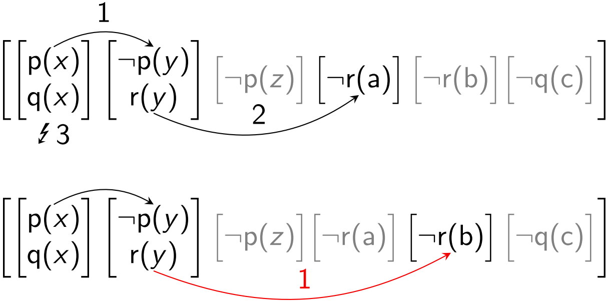
$$\left[\begin{bmatrix} \mathsf{p}(x) \\ \mathsf{q}(x) \end{bmatrix} \begin{bmatrix} \neg\mathsf{p}(y) \\ \mathsf{r}(y) \end{bmatrix} \begin{bmatrix} \neg\mathsf{p}(z) \end{bmatrix} \begin{bmatrix} \neg\mathsf{r}(\mathsf{a}) \end{bmatrix} \begin{bmatrix} \neg\mathsf{r}(\mathsf{b}) \end{bmatrix} \begin{bmatrix} \neg\mathsf{q}(\mathsf{c}) \end{bmatrix} \right] \tag{1}$$

Proof Search (Unrestricted Backtracking)

$$\left[\begin{bmatrix} p(x) \\ q(x) \end{bmatrix} \begin{bmatrix} \neg p(y) \\ r(y) \end{bmatrix} \begin{bmatrix} \neg p(z) \end{bmatrix} \begin{bmatrix} \neg r(a) \end{bmatrix} \begin{bmatrix} \neg r(b) \end{bmatrix} \begin{bmatrix} \neg q(c) \end{bmatrix}\right] \tag{1}$$

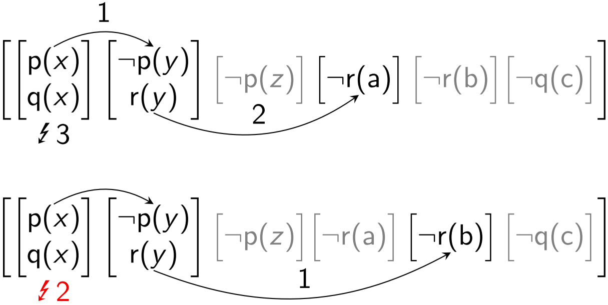
$$\left[\begin{bmatrix} p(x) \\ q(x) \end{bmatrix} \begin{bmatrix} \neg p(y) \\ r(y) \end{bmatrix} \begin{bmatrix} \neg p(z) \end{bmatrix} \begin{bmatrix} \neg r(a) \end{bmatrix} \begin{bmatrix} \neg r(b) \end{bmatrix} \begin{bmatrix} \neg q(c) \end{bmatrix}\right] \tag{2}$$

Proof Search (Unrestricted Backtracking)



(1)

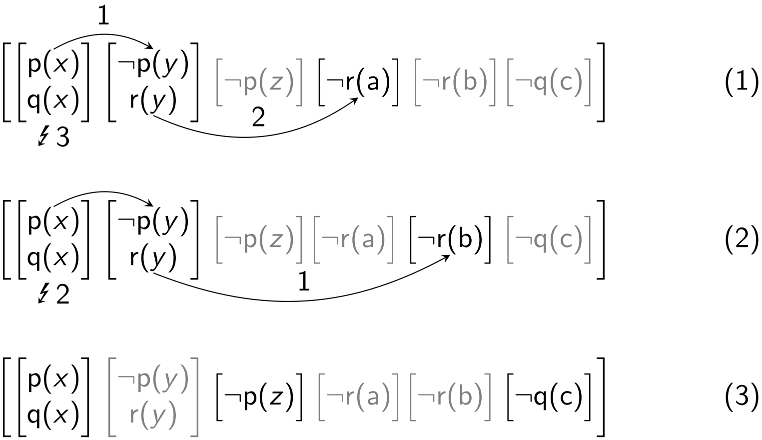
(2)



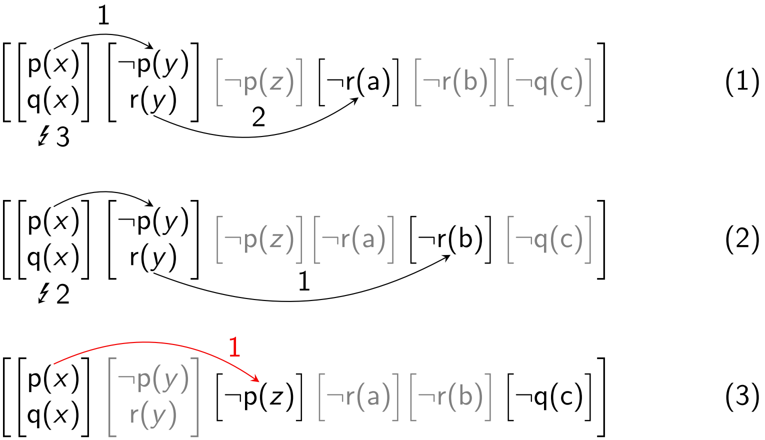
(1)

(2)

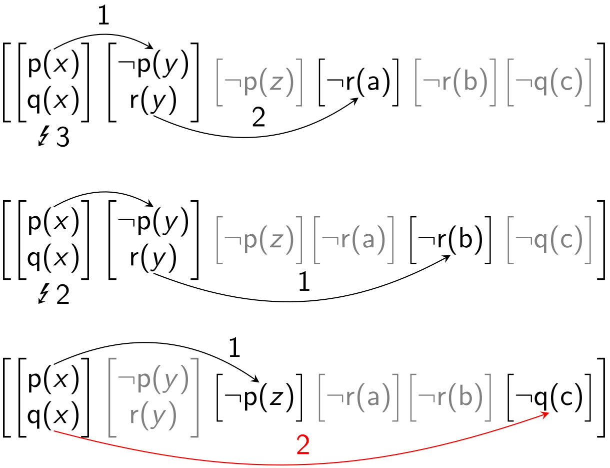
Proof Search (Unrestricted Backtracking)



Proof Search (Unrestricted Backtracking)



Proof Search (Unrestricted Backtracking)



(1)

(2)

(3)

Proof Search (Restricted Backtracking)

$$\left[\begin{bmatrix} p(x) \\ q(x) \end{bmatrix} \begin{bmatrix} \neg p(y) \\ r(y) \end{bmatrix} \begin{bmatrix} \neg p(z) \end{bmatrix} \begin{bmatrix} \neg r(a) \end{bmatrix} \begin{bmatrix} \neg r(b) \end{bmatrix} \begin{bmatrix} \neg q(c) \end{bmatrix} \right] \tag{1}$$

Proof Search (Restricted Backtracking)

$$\left[\begin{bmatrix} p(x) \\ q(x) \end{bmatrix} \overset{1}{\curvearrowright} \begin{bmatrix} \neg p(y) \\ r(y) \end{bmatrix} \begin{bmatrix} \neg p(z) \end{bmatrix} \begin{bmatrix} \neg r(a) \end{bmatrix} \begin{bmatrix} \neg r(b) \end{bmatrix} \begin{bmatrix} \neg q(c) \end{bmatrix} \right] \tag{1}$$

Proof Search (Restricted Backtracking)

$$\left[\begin{bmatrix} \mathrm{p}(x) \\ \mathrm{q}(x) \end{bmatrix} \begin{bmatrix} \neg\mathrm{p}(y) \\ \mathrm{r}(y) \end{bmatrix} \begin{bmatrix} \neg\mathrm{p}(z) \end{bmatrix} \begin{bmatrix} \neg\mathrm{r}(\mathrm{a}) \end{bmatrix} \begin{bmatrix} \neg\mathrm{r}(\mathrm{b}) \end{bmatrix} \begin{bmatrix} \neg\mathrm{q}(\mathrm{c}) \end{bmatrix} \right] \tag{1}$$

Proof Search (Restricted Backtracking)

$$\left[\begin{bmatrix} p(x) \\ q(x) \end{bmatrix} \begin{bmatrix} \neg p(y) \\ r(y) \end{bmatrix} \begin{bmatrix} \neg p(z) \end{bmatrix} \begin{bmatrix} \neg r(a) \end{bmatrix} \begin{bmatrix} \neg r(b) \end{bmatrix} \begin{bmatrix} \neg q(c) \end{bmatrix} \right] \tag{1}$$

Proof Search (Restricted Backtracking)

$$\left[\begin{bmatrix} p(x) \\ q(x) \end{bmatrix} \begin{bmatrix} \neg p(y) \\ r(y) \end{bmatrix} {\color{gray}\left[\neg p(z)\right]} \left[\neg r(a)\right] {\color{gray}\left[\neg r(b)\right]} {\color{gray}\left[\neg q(c)\right]}\right] \tag{1}$$

Restricted backtracking

- A literal is *solved* if it is connected to a literal L of a fresh clause copy C and all other literals $C \setminus L$ are solved
- Restricted backtracking does not consider alternative proofs for literals that have been solved
- Because $p(x)$ is solved, we cannot backtrack to find an alternative proof for it, thus proof search fails

Proof Search (Less Restricted Backtracking)

- Less restricted backtracking considers alternative proofs for literals that have been solved, but only if their root step is different

$$\left[\begin{bmatrix} \mathrm{p}(x) \\ \mathrm{q}(x) \end{bmatrix} \begin{bmatrix} \neg\mathrm{p}(y) \\ \mathrm{r}(y) \end{bmatrix} \begin{bmatrix} \neg\mathrm{p}(z) \end{bmatrix} \begin{bmatrix} \neg\mathrm{r}(\mathrm{a}) \end{bmatrix} \begin{bmatrix} \neg\mathrm{r}(\mathrm{b}) \end{bmatrix} \begin{bmatrix} \neg\mathrm{q}(\mathrm{c}) \end{bmatrix}\right] \tag{1}$$

Proof Search (Less Restricted Backtracking)

- Less restricted backtracking considers alternative proofs for literals that have been solved, but only if their root step is different

$$\left[\begin{bmatrix} \mathsf{p}(x) \\ \mathsf{q}(x) \end{bmatrix} \begin{bmatrix} \neg\mathsf{p}(y) \\ \mathsf{r}(y) \end{bmatrix} \begin{bmatrix} \neg\mathsf{p}(z) \end{bmatrix} \begin{bmatrix} \neg\mathsf{r}(\mathsf{a}) \end{bmatrix} \begin{bmatrix} \neg\mathsf{r}(\mathsf{b}) \end{bmatrix} \begin{bmatrix} \neg\mathsf{q}(\mathsf{c}) \end{bmatrix} \right] \tag{1}$$

Proof Search (Less Restricted Backtracking)

- Less restricted backtracking considers alternative proofs for literals that have been solved, but only if their root step is different

$$\left[\begin{bmatrix} \mathrm{p}(x) \\ \mathrm{q}(x) \end{bmatrix} \begin{bmatrix} \neg \mathrm{p}(y) \\ \mathrm{r}(y) \end{bmatrix} \begin{bmatrix} \neg \mathrm{p}(z) \end{bmatrix} \begin{bmatrix} \neg \mathrm{r}(\mathrm{a}) \end{bmatrix} \begin{bmatrix} \neg \mathrm{r}(\mathrm{b}) \end{bmatrix} \begin{bmatrix} \neg \mathrm{q}(\mathrm{c}) \end{bmatrix} \right] \tag{1}$$

- Less restricted backtracking considers alternative proofs for literals that have been solved, but only if their root step is different

$$\left[\begin{bmatrix} \mathrm{p}(x) \\ \mathrm{q}(x) \end{bmatrix} \begin{bmatrix} \neg\mathrm{p}(y) \\ \mathrm{r}(y) \end{bmatrix} \begin{bmatrix} \neg\mathrm{p}(z) \end{bmatrix} \begin{bmatrix} \neg\mathrm{r}(\mathrm{a}) \end{bmatrix} \begin{bmatrix} \neg\mathrm{r}(\mathrm{b}) \end{bmatrix} \begin{bmatrix} \neg\mathrm{q}(\mathrm{c}) \end{bmatrix} \right] \quad (1)$$

Proof Search (Less Restricted Backtracking)

- Less restricted backtracking considers alternative proofs for literals that have been solved, but only if their root step is different

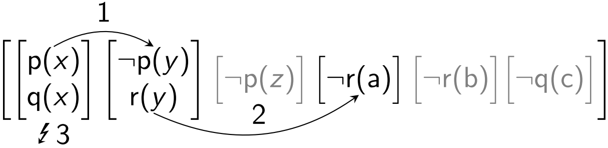
$$\left[\begin{bmatrix} p(x) \\ q(x) \end{bmatrix} \begin{bmatrix} \neg p(y) \\ r(y) \end{bmatrix} \begin{bmatrix} \neg p(z) \end{bmatrix} \begin{bmatrix} \neg r(a) \end{bmatrix} \begin{bmatrix} \neg r(b) \end{bmatrix} \begin{bmatrix} \neg q(c) \end{bmatrix} \right] \tag{1}$$

(arrows: 1 from $p(x)$ to $\neg p(y)$; 2 from $r(y)$ to $\neg r(a)$; $\lightning$ 3 at $q(x)$)

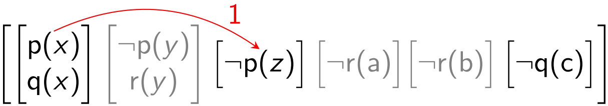
$$\left[\begin{bmatrix} p(x) \\ q(x) \end{bmatrix} \begin{bmatrix} \neg p(y) \\ r(y) \end{bmatrix} \begin{bmatrix} \neg p(z) \end{bmatrix} \begin{bmatrix} \neg r(a) \end{bmatrix} \begin{bmatrix} \neg r(b) \end{bmatrix} \begin{bmatrix} \neg q(c) \end{bmatrix} \right] \tag{3}$$

Proof Search (Less Restricted Backtracking)

- Less restricted backtracking considers alternative proofs for literals that have been solved, but only if their root step is different



(1)



(3)

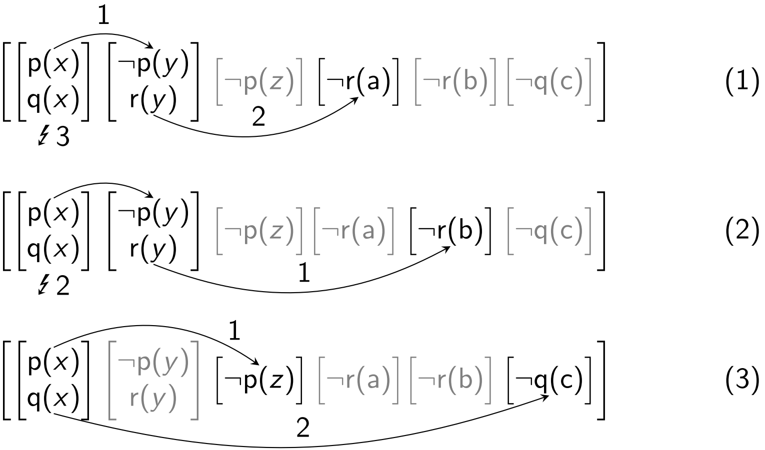
Proof Search (Less Restricted Backtracking)

- Less restricted backtracking considers alternative proofs for literals that have been solved, but only if their root step is different

$$\left[\begin{bmatrix} \mathrm{p}(x) \\ \mathrm{q}(x) \end{bmatrix} \begin{bmatrix} \neg\mathrm{p}(y) \\ \mathrm{r}(y) \end{bmatrix} \begin{bmatrix} \neg\mathrm{p}(z) \end{bmatrix} \begin{bmatrix} \neg\mathrm{r}(\mathrm{a}) \end{bmatrix} \begin{bmatrix} \neg\mathrm{r}(\mathrm{b}) \end{bmatrix} \begin{bmatrix} \neg\mathrm{q}(\mathrm{c}) \end{bmatrix}\right] \tag{1}$$

$$\left[\begin{bmatrix} \mathrm{p}(x) \\ \mathrm{q}(x) \end{bmatrix} \begin{bmatrix} \neg\mathrm{p}(y) \\ \mathrm{r}(y) \end{bmatrix} \begin{bmatrix} \neg\mathrm{p}(z) \end{bmatrix} \begin{bmatrix} \neg\mathrm{r}(\mathrm{a}) \end{bmatrix} \begin{bmatrix} \neg\mathrm{r}(\mathrm{b}) \end{bmatrix} \begin{bmatrix} \neg\mathrm{q}(\mathrm{c}) \end{bmatrix}\right] \tag{3}$$

Summary



Strategy	UB	RB	LRB
Stages	(1), (2), (3)	(1)	(1), (3)
Proof steps	7	3 ↯	4

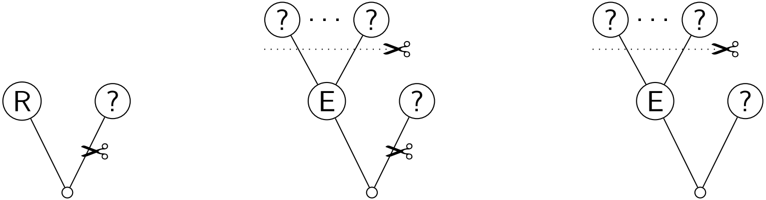
Section 3

A Zoo of Cuts

A Zoo of Cuts

Inclusive vs. Exclusive Cuts

- Inclusive cut discards all alternatives to solve a literal
- Exclusive cut discards all alternatives to solve a literal, *except for derivations starting with a different proof step*



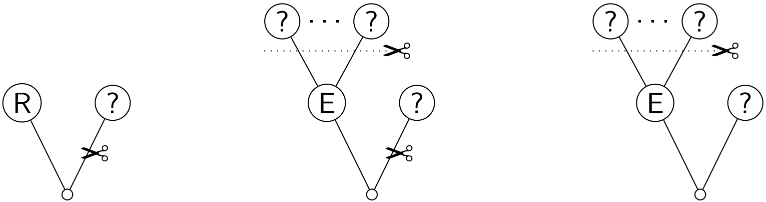
(a) Reduction (R). (b) Extension, inclusive (EI). (c) Extension, exclusive (EX).

Figure 1: Effect of different cuts on the tree of alternatives.

A Zoo of Cuts

Inclusive vs. Exclusive Cuts

- Inclusive cut discards all alternatives to solve a literal
- Exclusive cut discards all alternatives to solve a literal, *except for derivations starting with a different proof step*



(a) Reduction (R). (b) Extension, inclusive (EI). (c) Extension, exclusive (EX).

Figure 1: Effect of different cuts on the tree of alternatives.

It follows that cuts on reduction steps are always inclusive

Backtracking Strategies

A backtracking strategy is a set of cuts:

Backtracking	Cuts
Unrestricted	$\emptyset$ (None)
Restricted	{R, EI} (REI)
Less restricted	{R, EX} (REX)
Others	{R}, {EI}, {EX}

leanCoP's `cut` option corresponds to restricted backtracking, i.e. REI

Section 4

Implementation

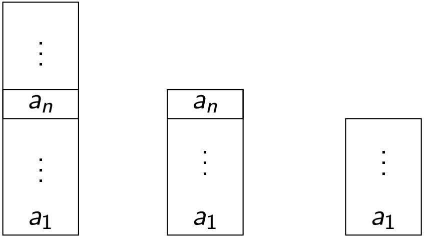
meanCoP

meanCoP

- connection prover for classical first-order logic with equality
- supports clausal and nonclausal proof search à la leanCoP/nanoCoP
- can perform precisely the same steps as leanCoP, for comparison
- includes a tiny proof checker and runs it before proof output
- written in Rust for high performance
- backed by the cop library, which provides building blocks for connection provers (formulas, terms, substitutions, backtracking . . .)

Get it on: https://github.com/01mf02/cop-rs

- meanCoP backtracks by using a stack of alternatives
- Whenever a literal is solved, the stack is shrunk



(a) No cut. (b) Exclusive. (c) Inclusive.

Figure 2: Effect of different cuts on the stack of alternatives.

Section 5

Evaluation

Datasets

Dataset	TPTP	bushy	chainy	Miz40	FS-top
Problems	7492	2078	2078	32524	27111

- TPTP 6.3.0: nonclausal first-order problems (`*+?.p`)
- bushy/chainy: from MPTP2078
- FS-top: translation to FOL of Flyspeck's HOL Light theorems

Timeout: 10 seconds/problem

Comparison of Cuts

Table 4: Number of solved problems.

Cut	TPTP	bushy	chainy	Miz40	FS-top
None	1731	546	208	9247	4038
R	1857	644	252	12965	4447
EI	1984	724	333	13853	4249
EX	2056	820	268	15507	4758
REI	1988	730	**341**	13562	4267
REX	**2126**	**850**	294	**16135**	**4994**

Table 5: Improvement of REX compared to REI.

Cut	TPTP	bushy	chainy	Miz40	FS-top
REX / REI	+6.9%	+16.4%	-13.8%	+19.0%	+17.0%

Comparison of Inferences

Table 6: Percentage of problems solved by C1 that are identically solved by C2.

C1	C2	TPTP	bushy	chainy	Miz40	FS-top
None	REX	84.5	66.5	89.4	77.8	81.0
None	REI	68.3	46.7	57.7	54.2	67.4
REX	REI	63.3	40.8	59.2	50.1	66.6

Table 7: Ratio between inferences taken by C1 and inferences taken by C2, for problems identically solved by C1 and C2.

C1	C2	TPTP	bushy	chainy	Miz40	FS-top
None	REX	4.4	37.0	9.9	37.4	19.8
None	REI	4.2	55.4	32.0	54.6	28.8
REX	REI	3.3	4.0	**8.4**	2.4	2.2

Comparison With Other Connection Provers

Table 8: Prover runtime in seconds for problems solved by leanCoP-REI.

Prover	TPTP	bushy	chainy	Miz40	FS-top
leanCoP-REI	1299.7	461.9	319.1	9308.7	2451.6
fleanCoP-REI	488.1	190.9	69.8	3845.6	657.2
meanCoP-REI	200.0	17.3	29.0	347.9	88.5

Table 9: Number of solved problems for different leanCoP implementations.

Prover	TPTP	bushy	chainy	Miz40	FS-top
leanCoP-REI	1673	606	182	11243	3664
fleanCoP-REI	1859	670	289	12204	3980
meanCoP-REI	1988	730	341	13562	4267

Comparison With Other Provers

Table 10: Number of solved problems by different provers.

Prover	TPTP	bushy	chainy	Miz40	FS-top
Vampire	4404	1253	656	30341	6358
E	3664	1167	287	26003	7382
Metis	1376	500	75	18519	3537
meanCoP-REI	1988	730	**341**	13562	4267
meanCoP-REX	2126	850	294	16135	4994

Comparison With Other Provers

Table 10: Number of solved problems by different provers.

Prover	TPTP	bushy	chainy	Miz40	FS-top
Vampire	4404	1253	656	30341	6358
E	3664	1167	287	26003	7382
Metis	1376	500	75	18519	3537
meanCoP-REI	1988	730	**341**	13562	4267
meanCoP-REX	2126	850	294	16135	4994

(Vampire 4.0 and E 2.0 were evaluated with strategy scheduling, giving them an advantage.)

Section 6

Conclusion

Conclusion

- Less Restricted Backtracking is a new backtracking strategy
- On most evaluated datasets, it clearly improves performance compared to restricted backtracking
- meanCoP is a connection prover with state-of-the-art performance

Open Question

Can we have less restricted backtracking in leanCoP?

Conclusion

- Less Restricted Backtracking is a new backtracking strategy
- On most evaluated datasets, it clearly improves performance compared to restricted backtracking
- meanCoP is a connection prover with state-of-the-art performance

Open Question

Can we have less restricted backtracking in leanCoP?

Thank you for your attention!