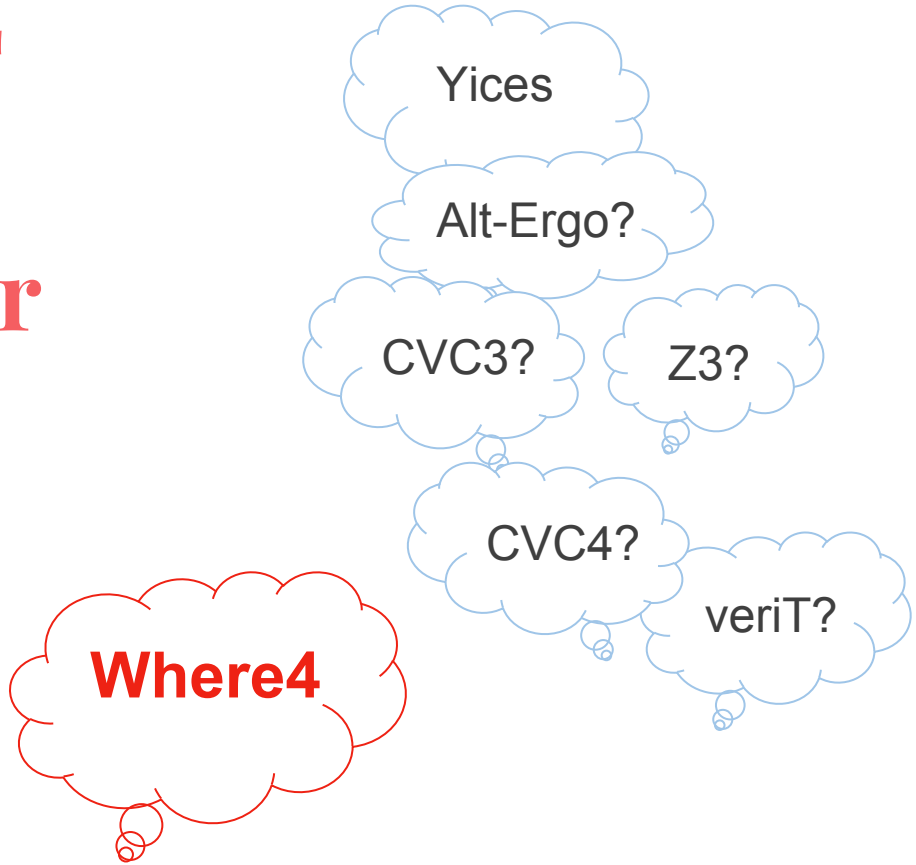
Reading the ingredients to predict the recipe

Andrew Healy

Postgraduate seminar 26-10-16

Predicting SMT Solver Performance for Software Verification



Where4 uses Why3

- Supports a range of input formats
  - .mlw (WhyML programming language)
  - .why (Why intermediate logic language)
  - .cnf (Dimacs format for Boolean formulas)
  - .p (TPTP FOL theorem prover library)
  - ...others via front-ends (Java, Ada, C)

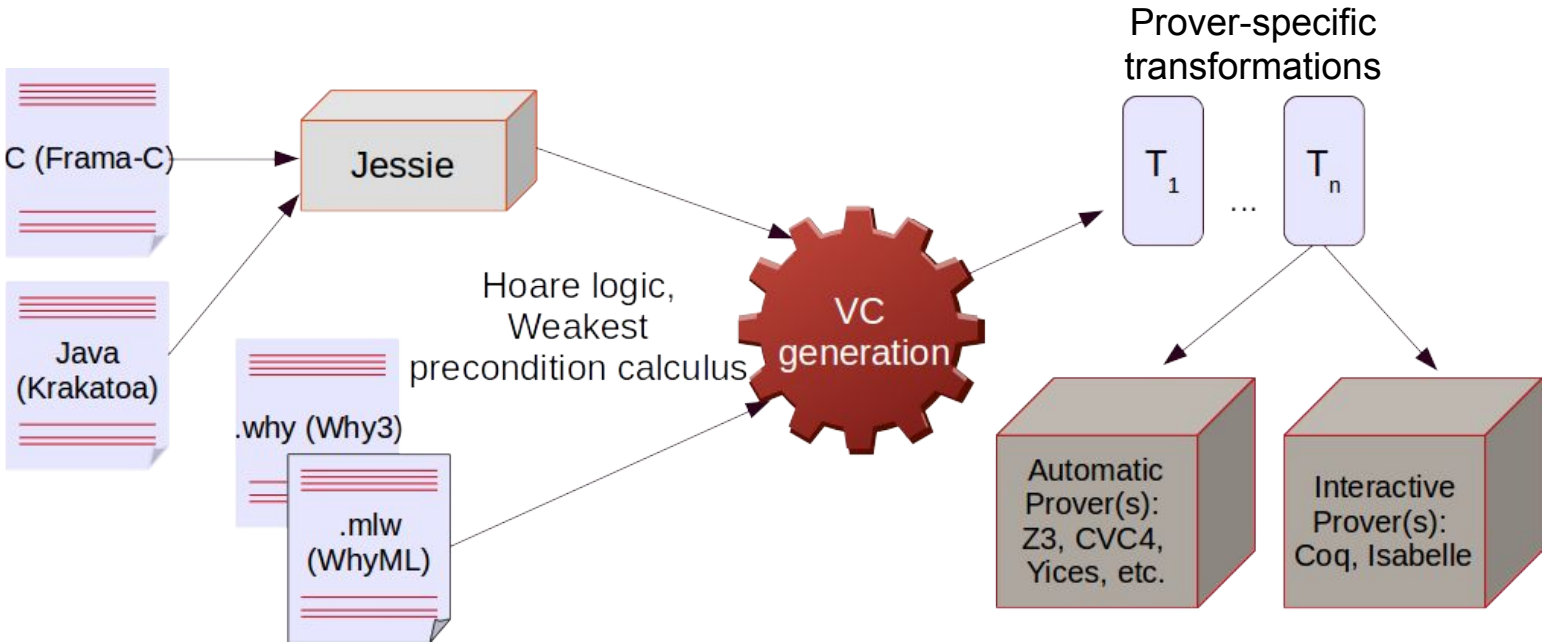
Where4 uses Why3

- Supports a range of input formats
  - .mlw (WhyML programming language)
  - .why (Why intermediate logic language)
  - .cnf (Dimacs format for Boolean formulas)
  - .p (TPTP FOL theorem prover library)
  - ...others via front-ends (Java, Ada, C)
- Supports a range of back-end tools
  - ATPs (automatic theorem provers)
  - ITPs (interactive theorem provers)
  - SMT (Satisfiability Modulo Theories - like extended SAT solvers)

Where4 uses Why3

- Supports a range of input formats
  - .mlw (WhyML programming language)
  - .why (Why intermediate logic language)
  - .cnf (Dimacs format for Boolean formulas)
  - .p (TPTP FOL theorem prover library)
  - ...others via front-ends (Java, Ada, C)
- Supports a range of back-end tools
  - ATPs (automatic theorem provers)
  - ITPs (interactive theorem provers)
  - SMT (Satisfiability Modulo Theories - like extended SAT solvers)
- Extensible architecture
  - OCaml API to create sessions, call solvers etc.
  - Users can write new drivers to control how solvers are called

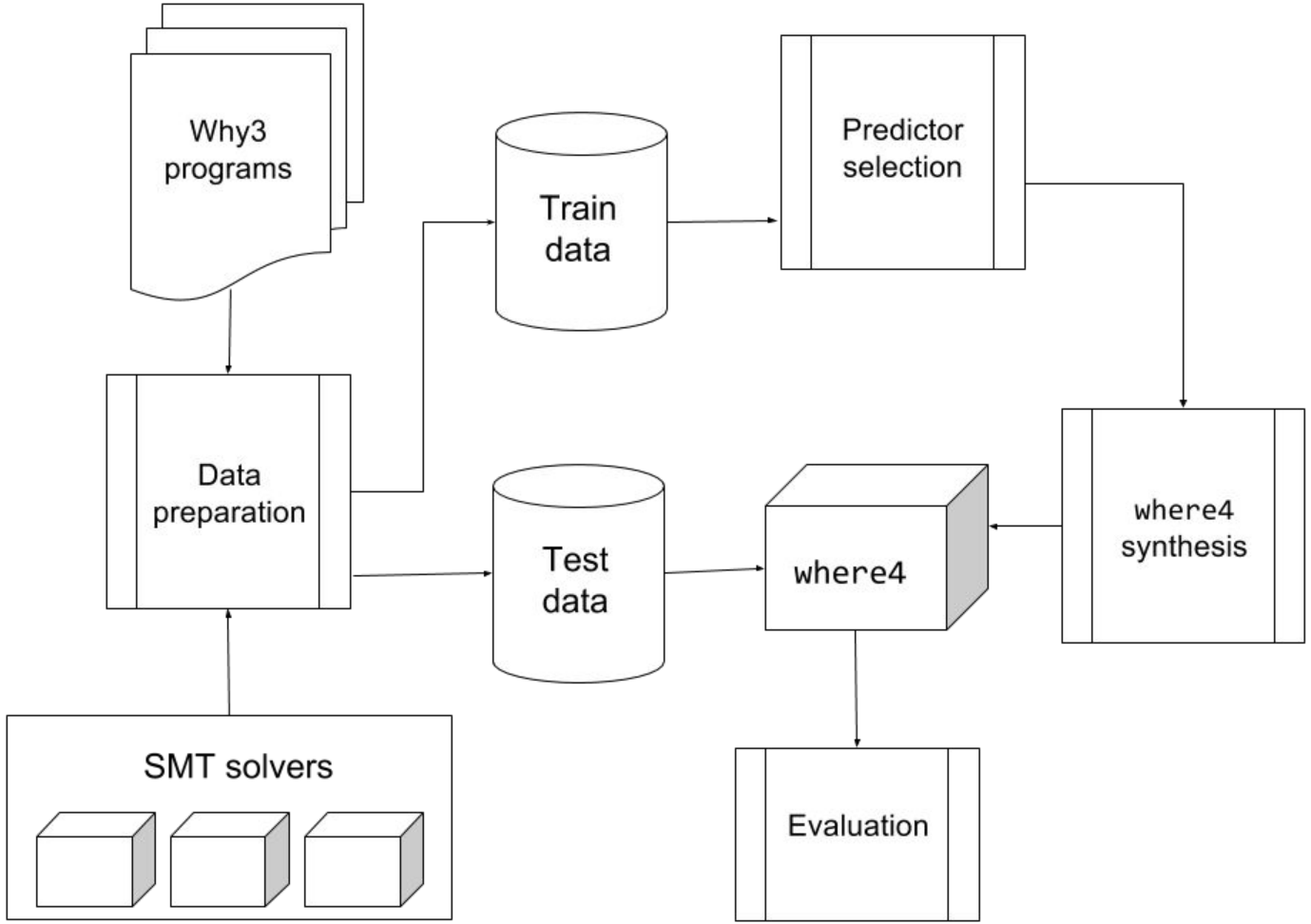
Where4 uses Why3



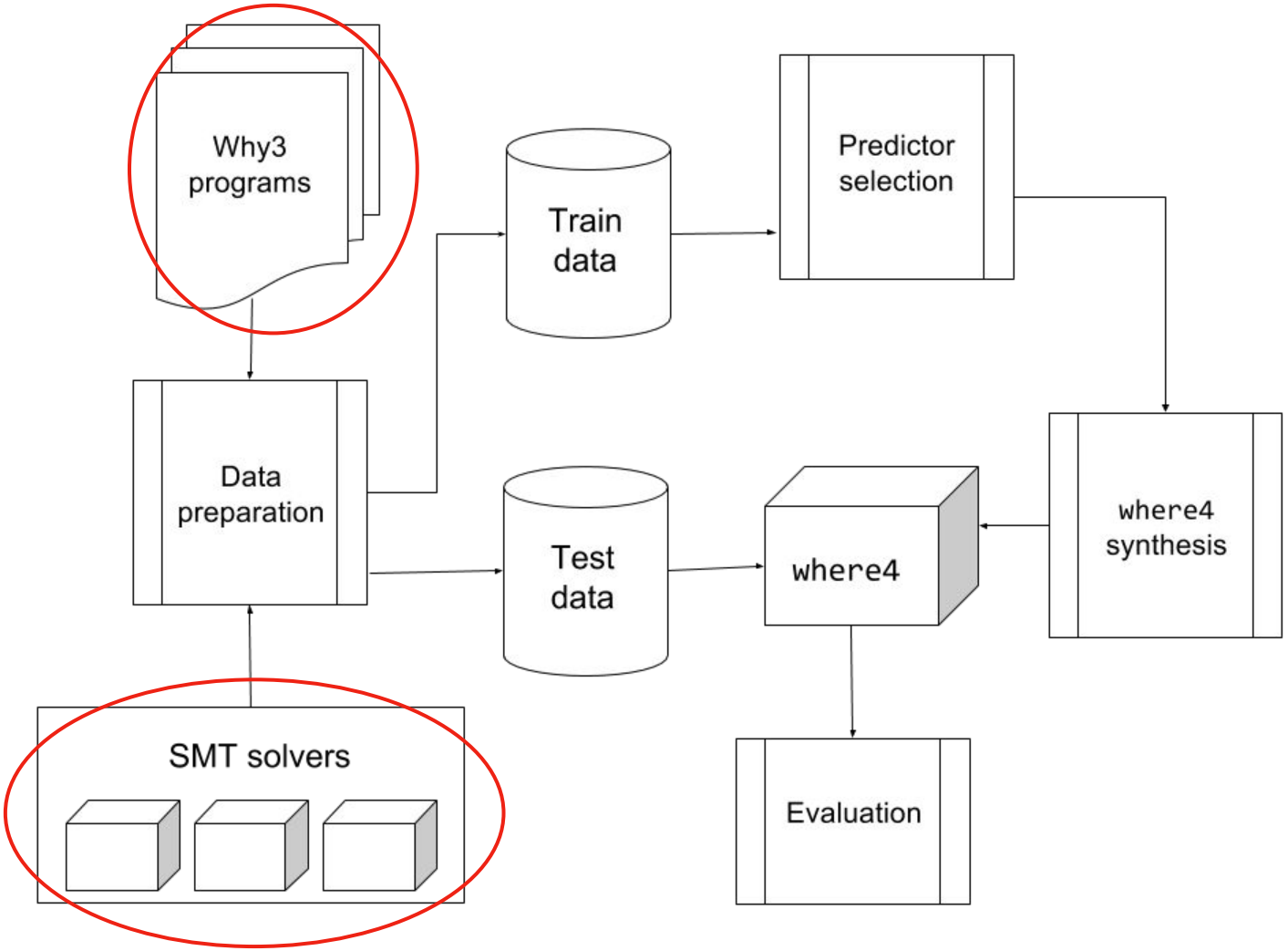
Portfolio solver motivation

	File			Theory			Goal		
	# proved	% proved	Avg time	# proved	% proved	Avg time	# proved	% proved	Avg time
Choose Single	**48**	**37.5%**	**1.90**	**190**	**63.8%**	**1.03**	**837**	**79.9%**	**0.42**
Alt-Ergo-0.95.2	25	19.5%	1.45	118	39.6%	0.77	568	54.2%	0.54
Alt-Ergo-1.01	34	26.6%	1.70	142	47.7%	0.79	632	60.3%	0.48
CVC3	19	14.8%	1.06	128	43.0%	0.65	597	57.0%	0.49
CVC4	19	14.8%	1.09	117	39.3%	0.51	612	58.4%	0.37
veriT	5	4.0%	0.12	79	26.5%	0.20	333	31.8%	0.26
Yices	14	10.9%	0.53	102	34.2%	0.22	368	35.1%	0.22
Z3-4.3.2	25	19.5%	0.56	128	43.0%	0.36	488	46.6%	0.38
Z3-4.4.1	26	20.3%	0.58	130	43.6%	0.40	581	55.4%	0.35

Developing Where4



Developing Where4



Data & Tool Selection

Why3 programs

128 programs from the Why3 distribution example directory

Mostly completed solutions to SV competitions (verifyThis, COST, VACID-0)

298 theories, 1048 goals

Data & Tool Selection

Why3 programs

128 programs from the Why3 distribution example directory

Mostly completed solutions to SV competitions (verifyThis, COST, VACID-0)

298 theories, 1048 goals

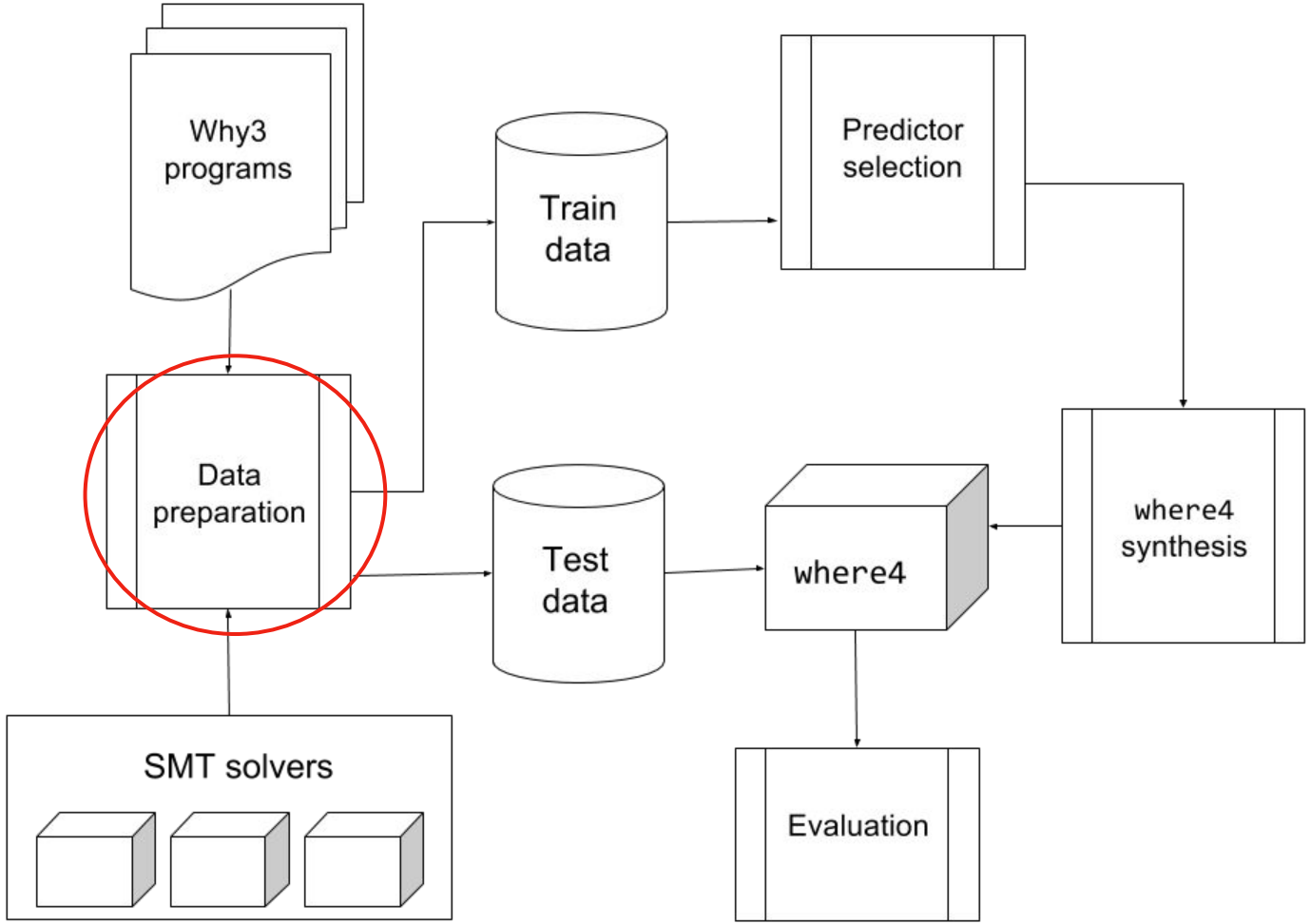
SMT solvers

- Z3
- Alt-Ergo
- CVC3
- CVC4
- veriT
- Yices

Current, widely used for SV

Two versions of Z3 & Alt-Ergo (more flexible: can reflect the user's local installation)

Developing Where4



Data Preparation

Accurately measure solver responses

Record not just the solver response:

- Valid
- Invalid
- Unknown
- Timeout
- Failure

But also the time taken to return this response

Data Preparation

Accurately measure solver responses

Record not just the solver response:

- Valid
- Invalid
- Unknown
- Timeout
- Failure

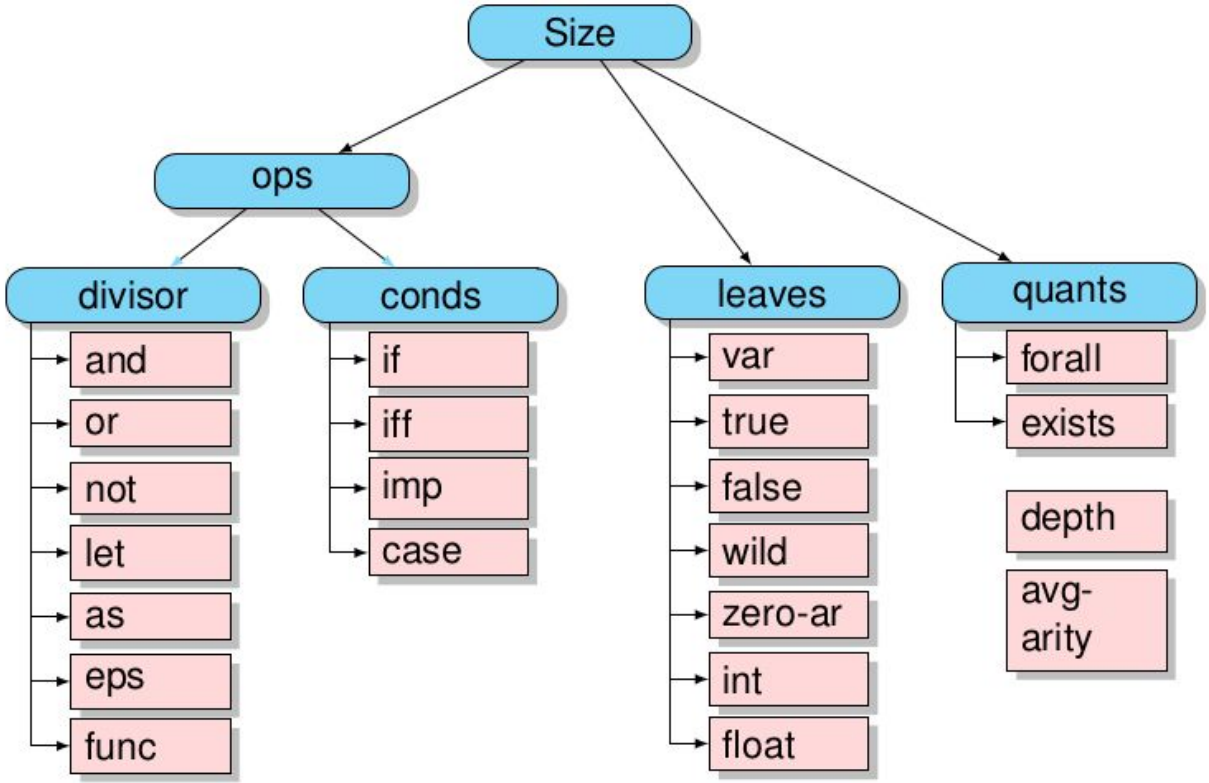
But also the time taken to return this response

Extract syntactic features from goals

Traverse the abstract syntax tree for each formula sent to the solver.

Count the depth of the tree, number of leaves, number of each types met during traversal.

Data Preparation

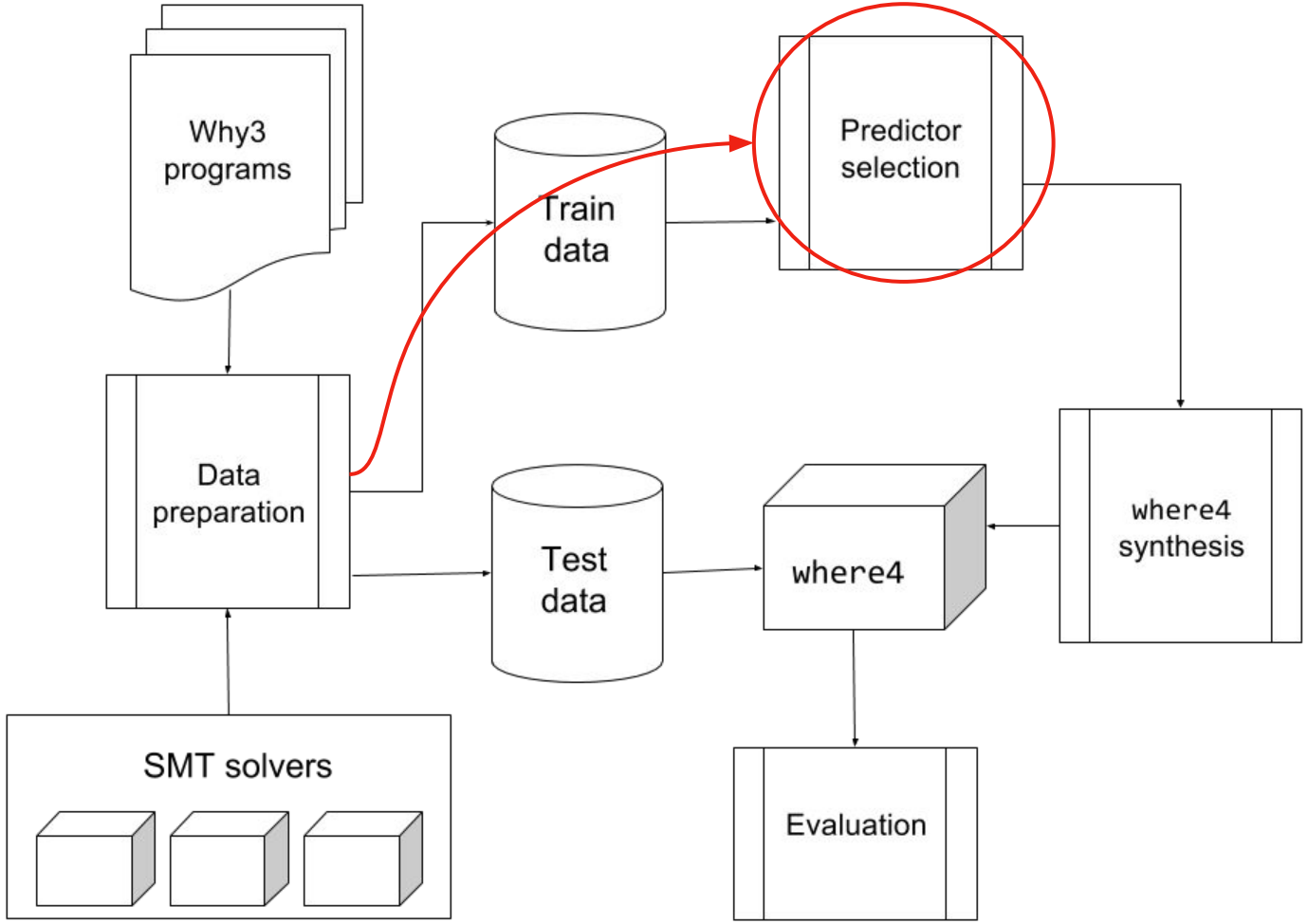


Data Preparation

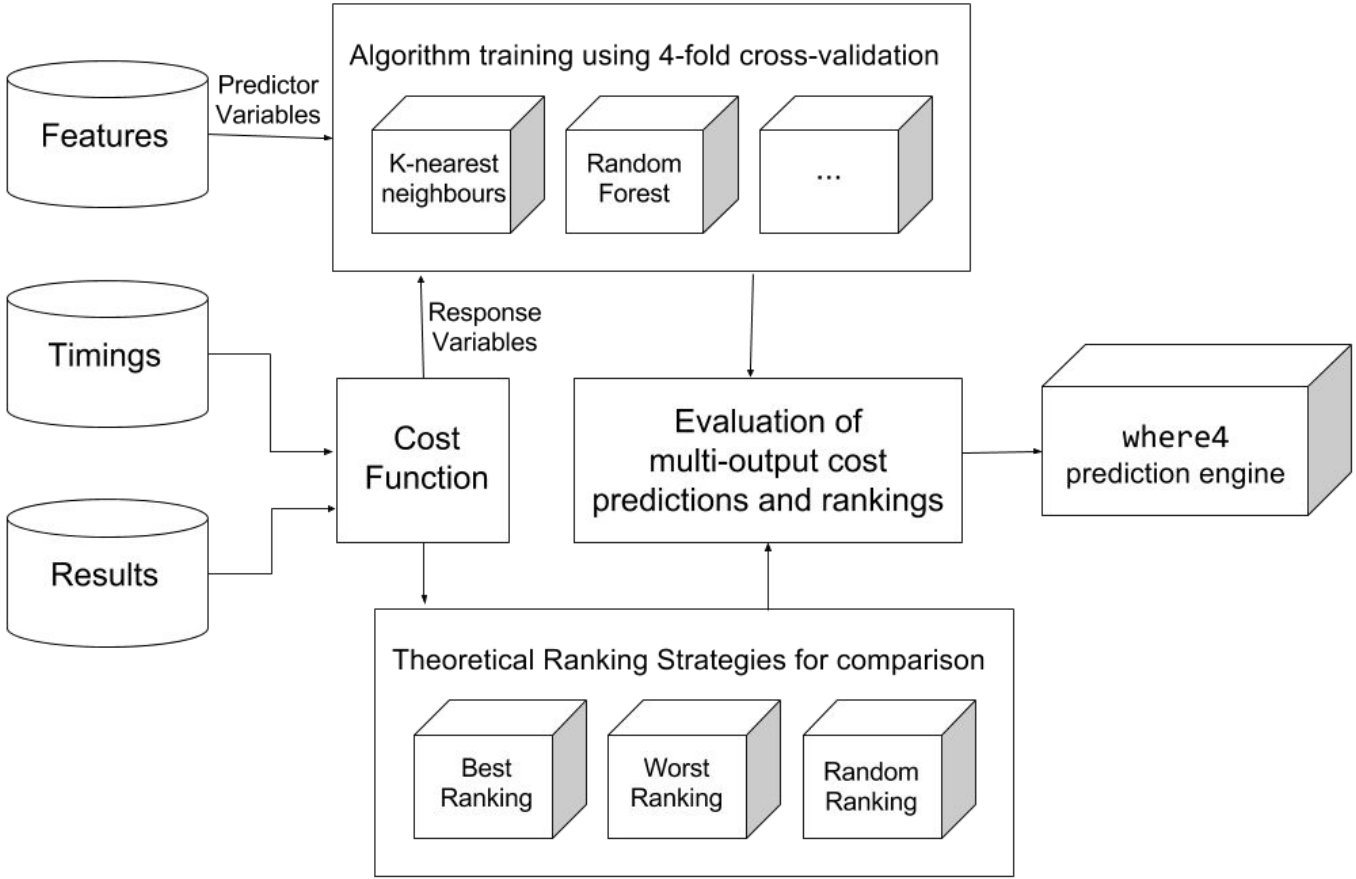

```
"and": 112.0,
"false": 0.0,
"exists": 6.0,
"int": 37.0,
"float": 0.0,
"as": 0.0,
"n_quants": 39.0,
"zero_ar": 0.0,
"if": 4.0,
"goal": "WP_parameter find",
"n_branches": 443.0,
"iff": 0.0,
"n_ops": 408.0,
"var": 406.0,
"size": 890.0,
```

```
"avg_op_arity": 2.200495,
"theory": "Algo65",
"n_preds": 4.0,
"forall": 33.0,
"let": 5.0,
"func": 254.0,
"not": 0.0,
"true": 0.0,
"case": 0.0,
"divisor": 404.0,
"eps": 0.0,
"depth": 34.0,
"wild": 0.0,
"or": 0.0,
"impl": 33.0
```

Developing Where4



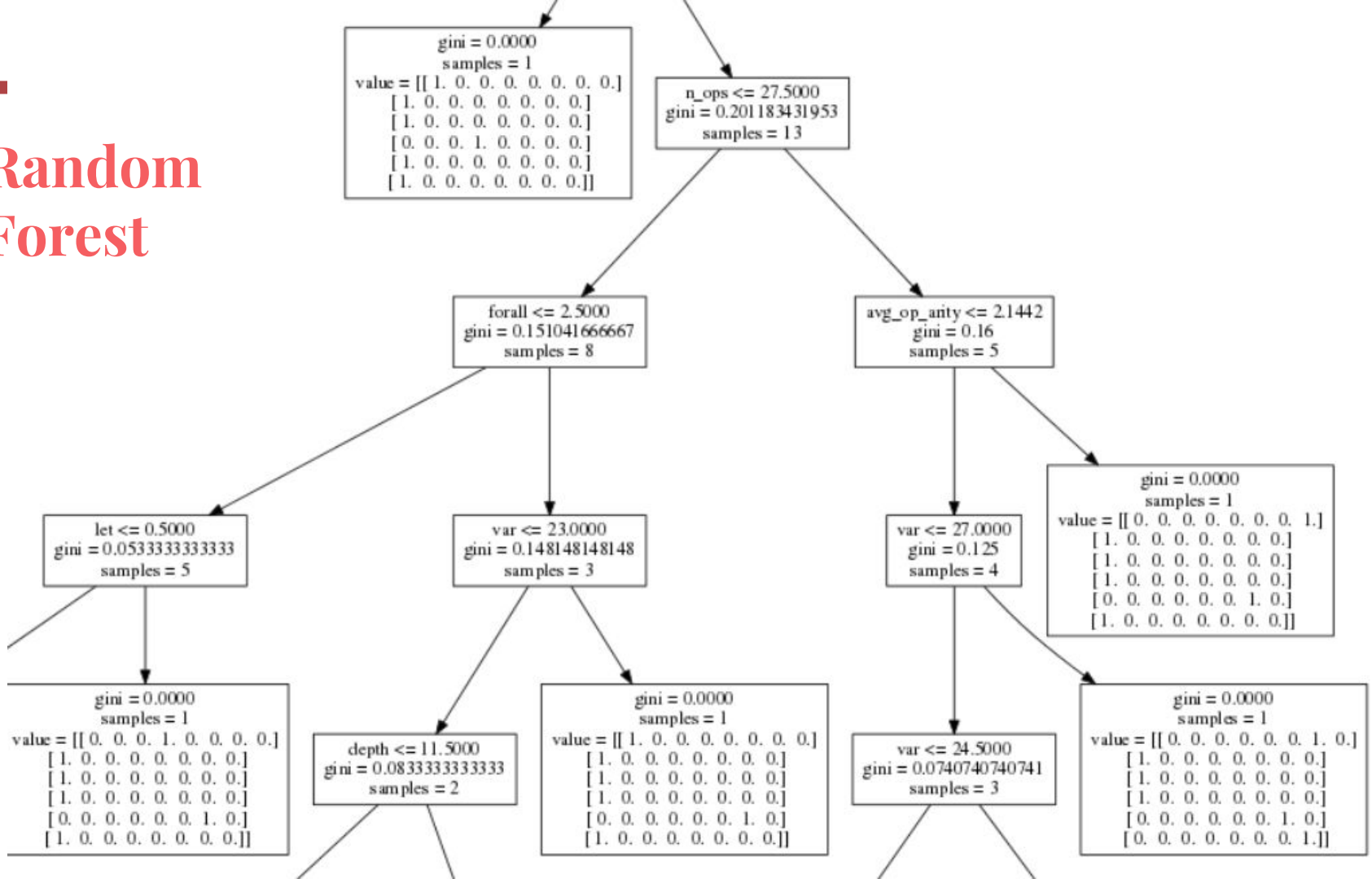
Predictor Selection



Predictor Selection

	Time (secs)	nDCG	R^2	MAE	Reg. error
Best	12.63	1.00	-	0.00	0.00
Random	19.06	0.36	-	2.63	50.77
Worst	30.26	0.00	-	4.00	94.65
Random Forest	15.02	0.48	**0.28**	2.08	**38.91**
Random Forest (discretised)	**14.92**	0.48	-0.18	2.13	39.19
Decision Tree	15.80	0.50	0.11	2.06	43.12
K-Nearest Neighbours	15.93	**0.53**	0.16	**2.00**	43.41
Support Vector Regressor	15.57	0.47	0.14	2.26	47.45
Linear Regression	15.17	0.42	-0.16	2.45	49.25
Ridge	15.11	0.42	-0.15	2.45	49.09

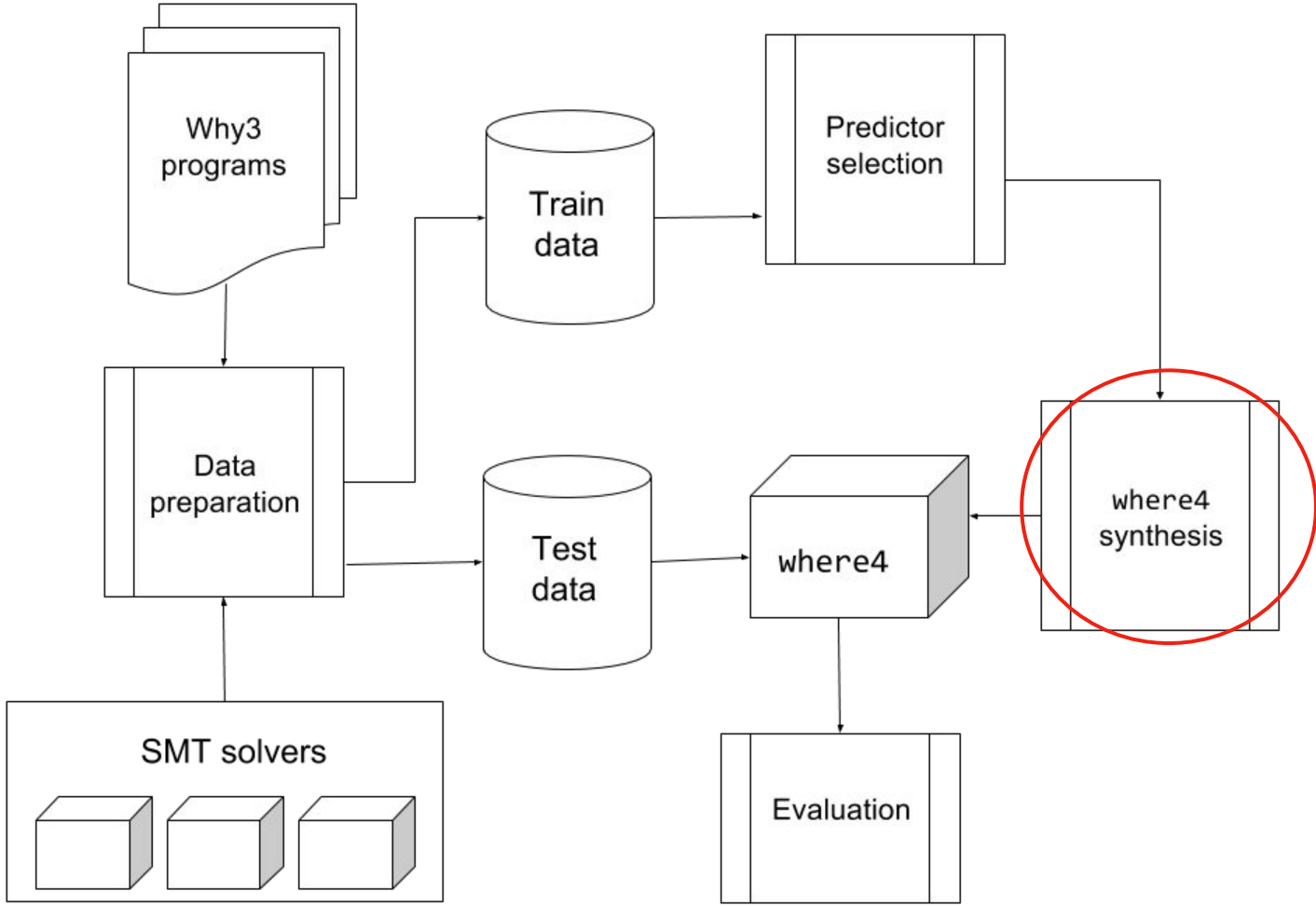
Random Forest



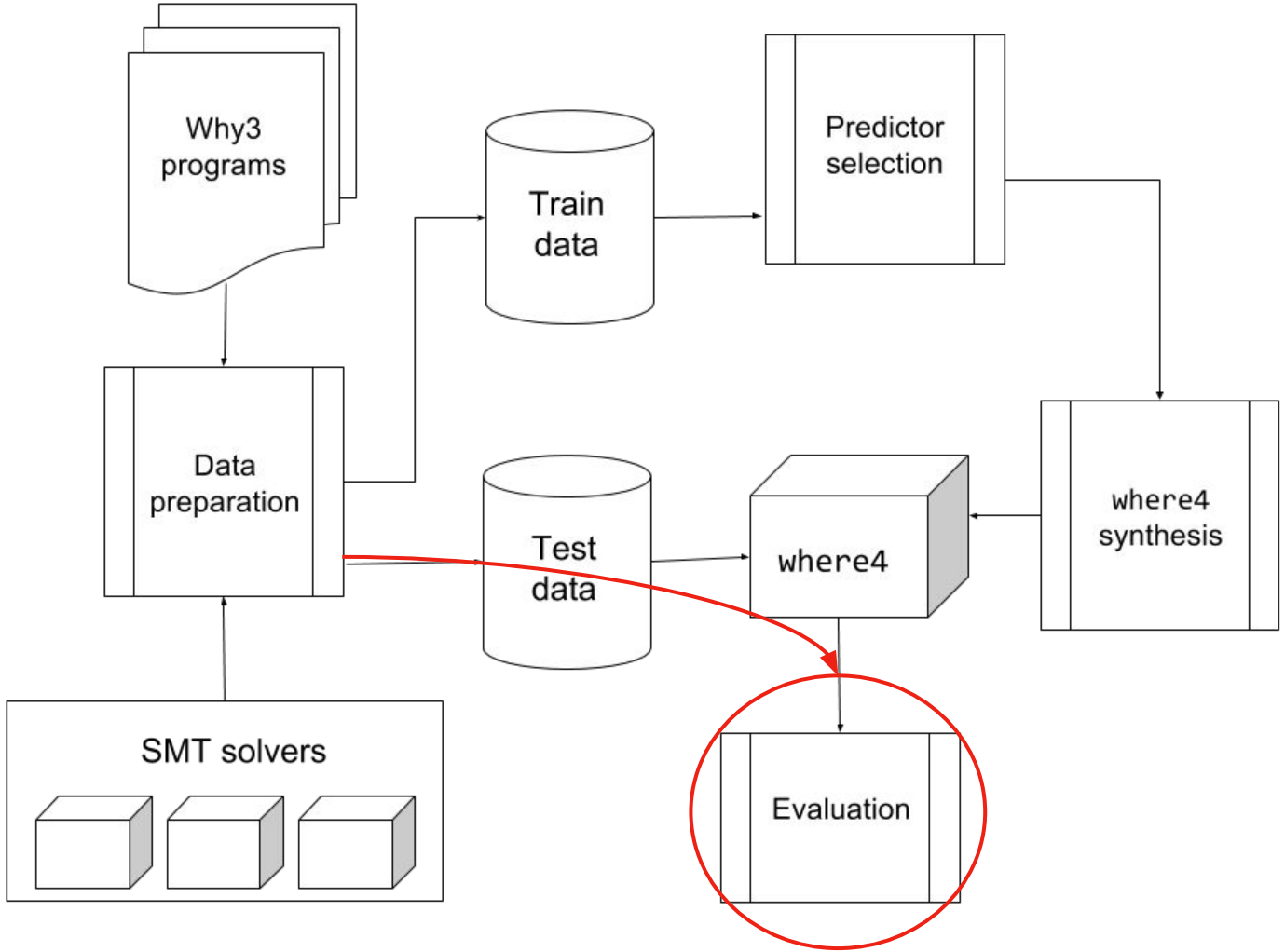
Random Forest

Feature	Importance (%)	Feature	Importance (%)
func	19.94	n-ops	2.90
avg-op-arity	10.55	divisor	2.84
int	8.68	not	1.18
and	5.83	if	1.15
forall	5.75	case	0.97
depth	5.44	wild	0.95
impl	4.98	false	0.38
var	4.98	iff	0.29
n-quants	4.64	or	0.28
size	4.61	true	0.12
let	3.91	eps	0.09
n-branches	3.40	exists	0.04
n-preds	3.12	float	0.03
zero-ar	3.00	as	0.00

Developing Where4



Developing Where4



Evaluation

1. How does Where4 compare to using an individual solver?

2. How does Where4 compare to using a theoretical ranking strategy?

3. What is the time overhead of using Where4?

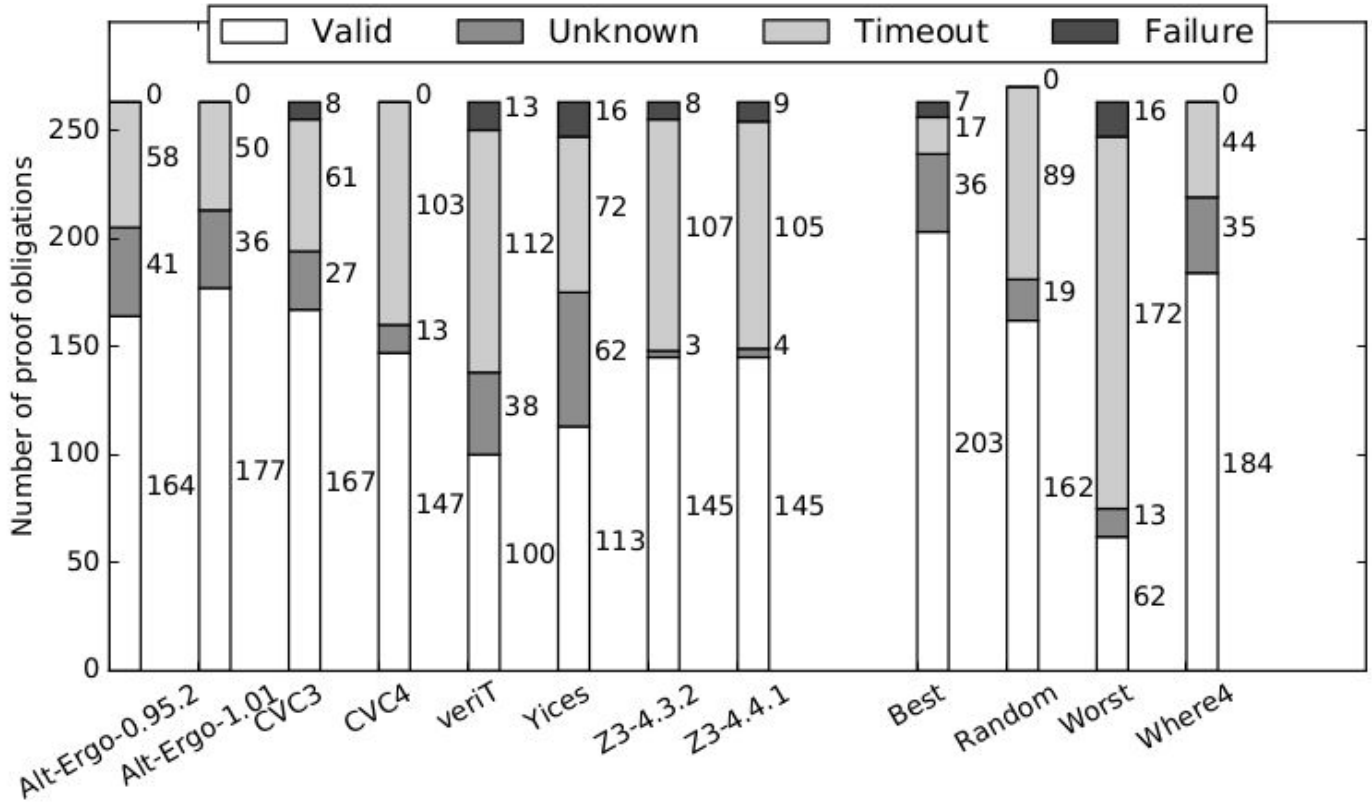
Evaluation

1. How does Where4 compare to using an individual solver?

	File			Theory			Goal		
	# proved	% proved	Avg time	# proved	% proved	Avg time	# proved	% proved	Avg time
Where4	11	34.4%	1.39	44	57.1%	0.99	203	77.2%	1.98
Best Rank.			0.25			0.28			0.37
Random Rank.			4.19			4.02			5.70
Worst Rank.			14.71			13.58			18.35
Alt-Ergo-0.95.2	8	25.0%	0.78	37	48.1%	0.26	164	62.4%	0.34
Alt-Ergo-1.01	10	31.3%	1.07	39	50.6%	0.26	177	67.3%	0.33
CVC3	5	15.6%	0.39	36	46.8%	0.21	167	63.5%	0.38
CVC4	4	12.5%	0.56	32	41.6%	0.21	147	55.9%	0.35
veriT	2	6.3%	0.12	24	31.2%	0.12	100	38.0%	0.27
Yices	4	12.5%	0.32	32	41.6%	0.15	113	43.0%	0.18
Z3-4.3.2	6	18.8%	0.46	31	40.3%	0.20	145	55.1%	0.37
Z3-4.4.1	6	18.8%	0.56	31	40.3%	0.23	145	55.1%	0.38

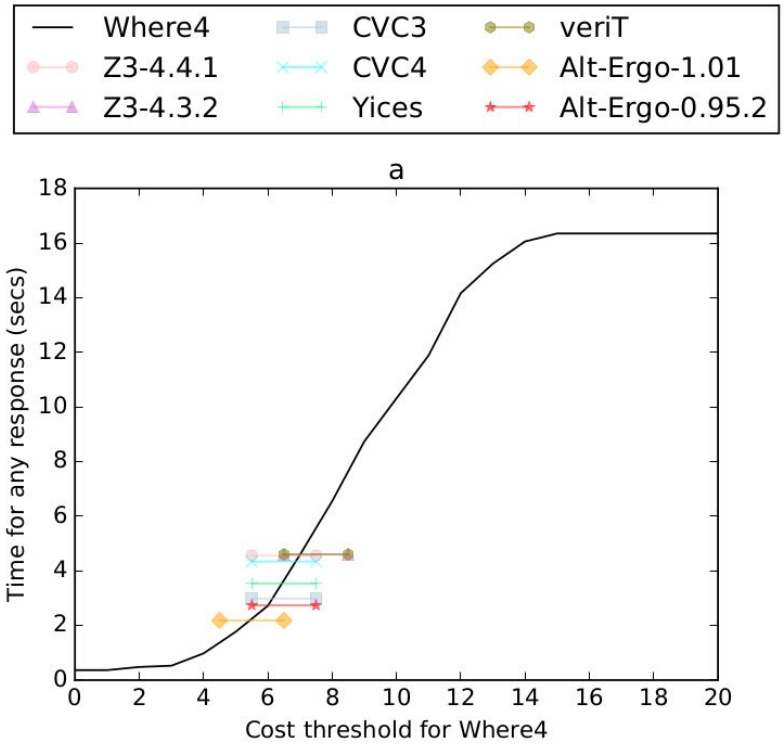
Evaluation

1. How does Where4 compare to using an individual solver?



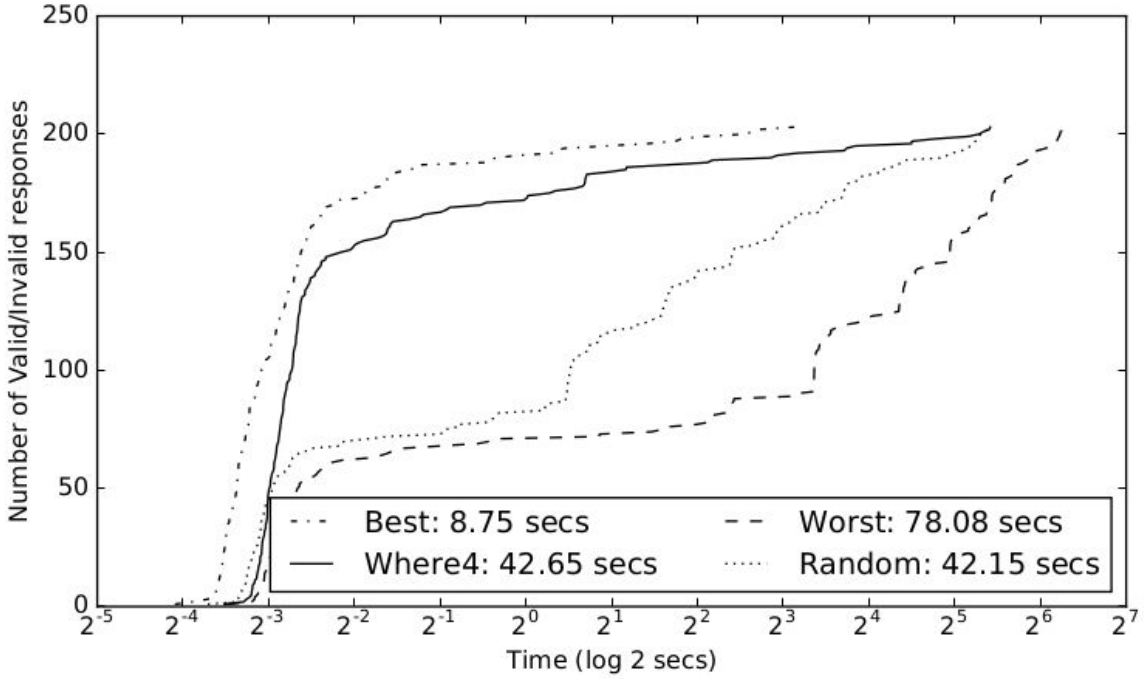
Evaluation

1. How does Where4 compare to using an individual solver?



Evaluation

How does Where4 compare to using a theoretical ranking strategy?



Evaluation

What is the time overhead of using Where4?

About 0.5 seconds per file for predictions

Equivalent to an average of 0.05 seconds per goal

Contributions

- Comparative analysis of 8 SMT solvers' performance on a large dataset
- A suite of metrics to characterise goals for the Why3 system
- Evaluation of 6 predictions models' suitability for our multi-output regression task
- Command-line tool to predict the ranking of solvers using a Random Forest model
- Command-line portfolio solver and Why3 "imitation solver"

Thanks!