Fingerprint Indexing for Paramodulation

Stephan Schulz
schulz@eprover.org

Core problem

Given:

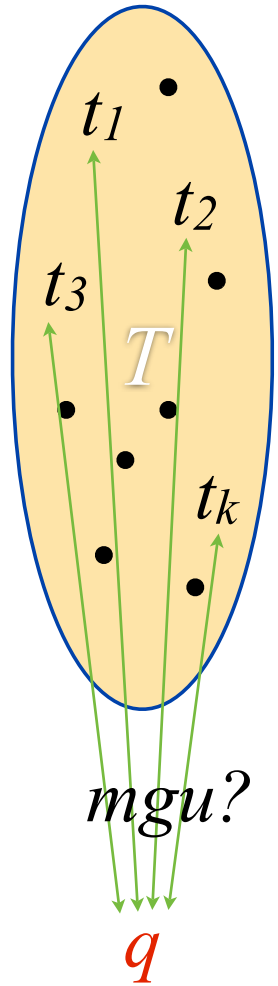
- A set of first-order terms T
- A query term q

Find (quickly):

- All $t \in T$ such that $mgu(q, t)$ exists

Side condition:

- Repeated queries with different q
- T is big ($\gg$ 10000 terms)
- T evolves
  - Frequent additions
  - Infrequent deletions



Properties of Unification

Observations

Unification creates a common instance of the two terms

Instantiation never removes positions from a term

Instantiation never changes a function symbol in a term

Instantiation at most adds positions below variables

Properties of Unification

Observations

Unification creates a common instance of the two terms

Instantiation never removes positions from a term

Instantiation never changes a function symbol in a term

Instantiation at most adds positions below variables

Idea

Formulate necessary conditions for unification on (potential) position in terms

Term Sampling

Consider t at (potential) position $p \in \mathbb{N}^*$

- $\mathcal{N}$: p does not exist in t or any instance
- $\mathcal{B}$: p is below a variable position in t
- $\mathcal{A}$: p denotes the occurrence of a variable in t
- f: t_p starts with function symbol f

Sampling function:

$$fps : Term(F, V) \times \mathbb{N}^* \rightarrow F \uplus \{\mathcal{A}, \mathcal{B}, \mathcal{N}\}$$

$$fps(t, p) = \begin{cases} \mathcal{A} & \text{if } p \in O(t),\ t|_p \in V \\ head(t|_p) & \text{if } p \in O(t),\ t|_p \notin V \\ \mathcal{B} & \text{if } p = q.r,\ q \in O(t) \text{ and } t|_q \in V \\ \mathcal{N} & \text{otherwise} \end{cases}$$

Sample Compatibility

Across: $fps(s, p)$

Down: $fps(t, p)$

Entry: Possiby unifiable?

	f_1	f_2	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{N}$
f_1	Y	N	Y	Y	N
f_2	N	Y	Y	Y	N
$\mathcal{A}$	Y	Y	Y	Y	N
$\mathcal{B}$	Y	Y	Y	Y	Y
$\mathcal{N}$	N	N	N	Y	Y

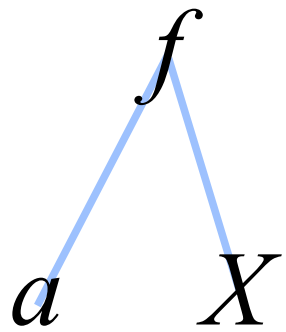
Sample Compatibility

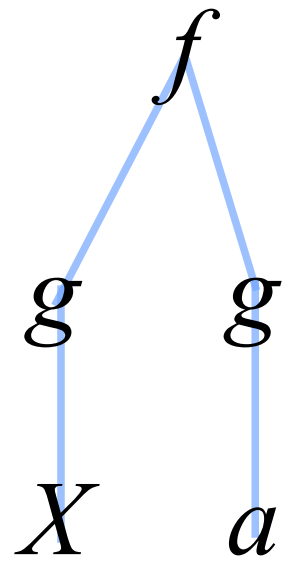
Across: $fps(s, p)$

Down: $fps(t, p)$

Entry: Possiby unifiable?

	f_1	f_2	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{N}$
f_1	Y	N	Y	Y	N
f_2	N	Y	Y	Y	N
$\mathcal{A}$	Y	Y	Y	Y	N
$\mathcal{B}$	Y	Y	Y	Y	Y
$\mathcal{N}$	N	N	N	Y	Y





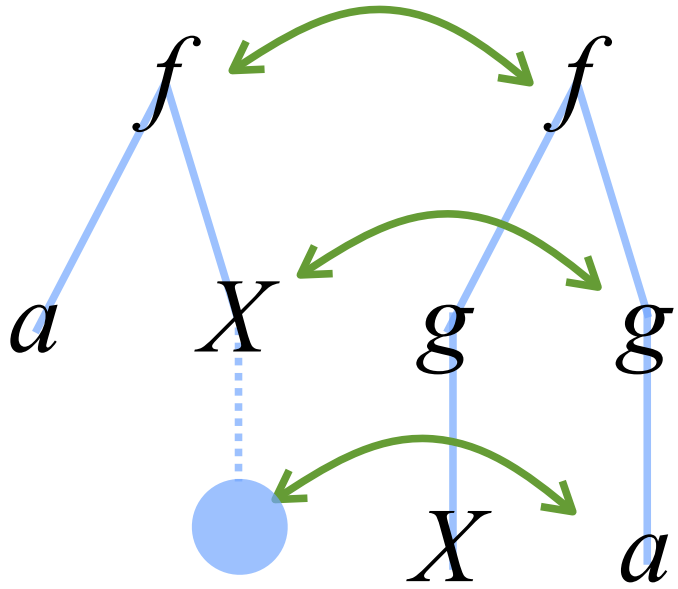
Sample Compatibility

Across: $fps(s, p)$

Down: $fps(t, p)$

Entry: Possiby unifiable?

	f_1	f_2	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{N}$
f_1	Y	N	Y	Y	N
f_2	N	Y	Y	Y	N
$\mathcal{A}$	Y	Y	Y	Y	N
$\mathcal{B}$	Y	Y	Y	Y	Y
$\mathcal{N}$	N	N	N	Y	Y



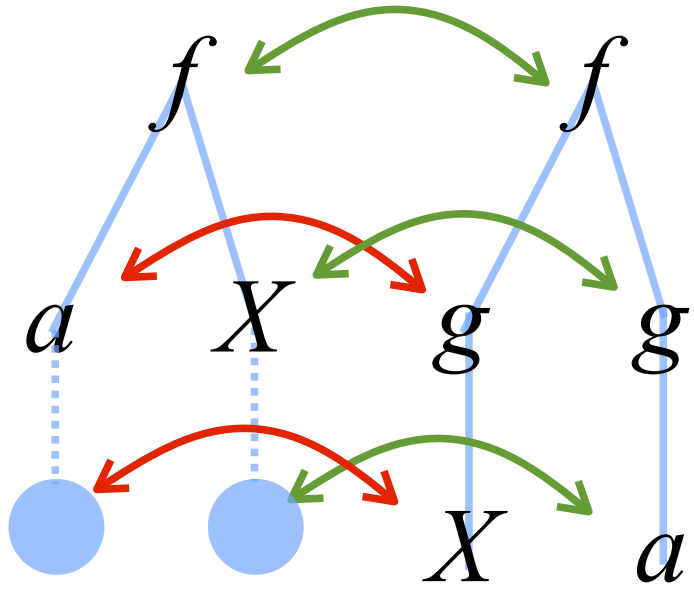
Sample Compatibility

Across: $fps(s, p)$

Down: $fps(t, p)$

Entry: Possiby unifiable?

	f_1	f_2	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{N}$
f_1	Y	N	Y	Y	N
f_2	N	Y	Y	Y	N
$\mathcal{A}$	Y	Y	Y	Y	N
$\mathcal{B}$	Y	Y	Y	Y	Y
$\mathcal{N}$	N	N	N	Y	Y



Fingerprint Indexing

Fingerprint: Fixed length vector of term samples

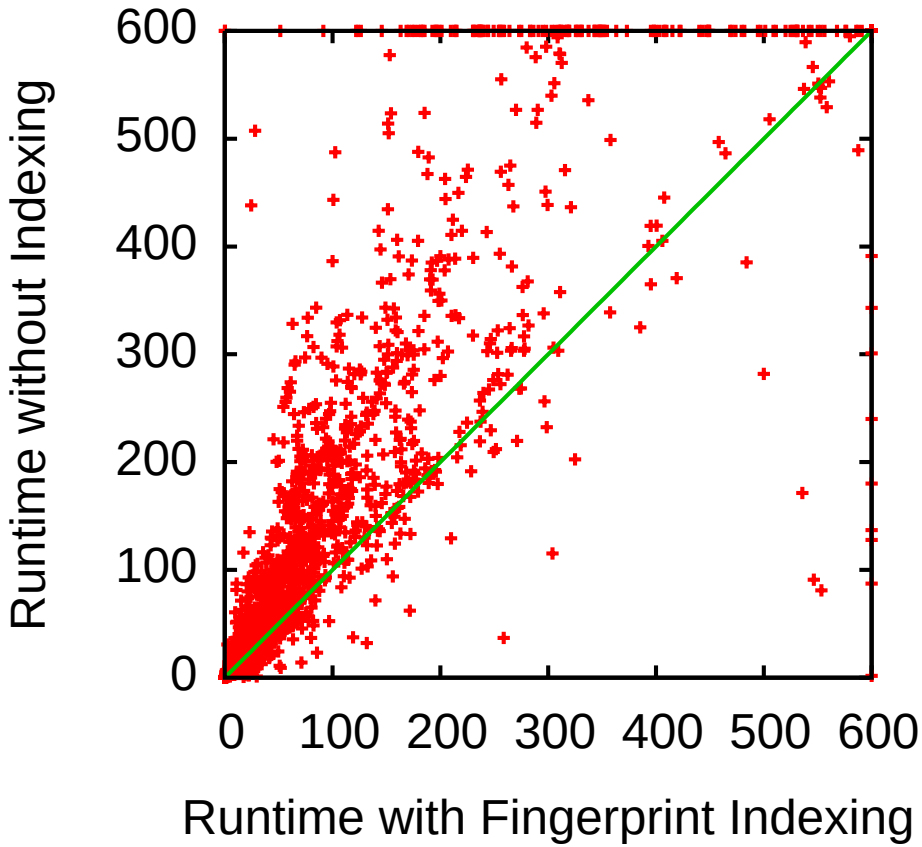
Fingerprints are organized in a trie

- Every term is represented only once in the index
- Index is compact (fingerprint length $\ll 10$)

Retrieval: Follow all compatible branches

- For identity only one choice
- For unification use preceding table
- For matching use alternative tables

Performance (very preliminary)



- E 1.2 (pre-release)
- TPTP 4.0.1
  - ≈14000 problems
  - ≈8500 solutions
- Abstract runtimes (normalized)
- 5-sample Fingerprint

Sales Pitch

Fingerprint indexing inspired by

- Top symbol hashing
- Path indexing
- Feature vector indexing

Retrieval operations

- Identity
- Unification
- Matching (from and into)

Properties

- Non-perfect
- Compact data structures
- Simple to implement
- Efficient in practice


The End!