Term-Indexing

First-order terms

Alexander Maringele

January 27th, 2016

References

R. Sekar, I. V. Ramakrishnan, and Andrei Voronkov, *Term indexing*, Handbook of Automated Reasoning (Alan Robinson and Andrei Voronkov, eds.), Elsevier Science Publishers B. V., Amsterdam, The Netherlands, 2001, pp. 1853–1964.

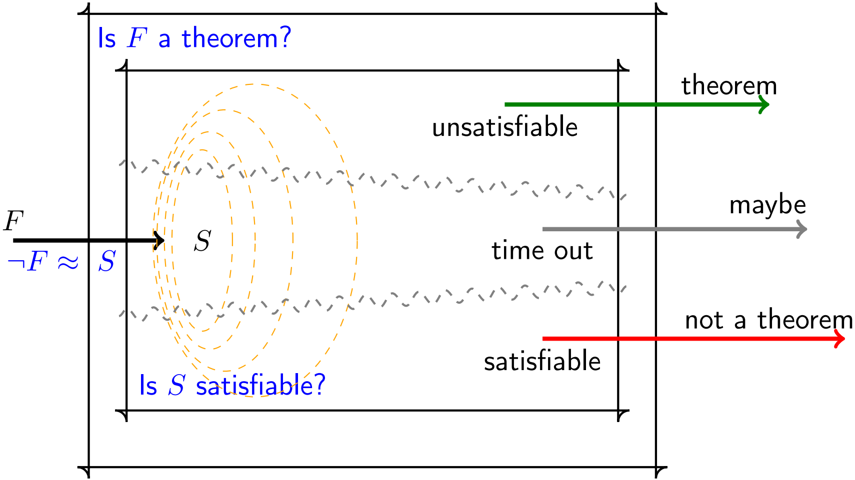
Outline

Notation

Example

$$\Sigma = (\mathcal{V}, \mathcal{F}, \mathcal{P}) \qquad \text{signature}$$
$$\mathcal{V} = \{x, y, z, \ldots\} \qquad \text{variables}$$
$$\mathcal{F} = \{\ \mathsf{a}, \mathsf{f}, \mathsf{g}, \mathsf{h}, \ldots\} \qquad \text{function symbols}$$
$$\mathcal{P} = \{\mathsf{P}, \mathsf{Q}, \ldots\} \qquad \text{predicate symbols}$$
$$\mathcal{T} = \mathcal{V} \cup \{\ f(t_1, \ldots, t_n) \mid f \in \mathcal{F}, t_i \in \mathcal{T}\} \qquad \text{terms}$$
$$\mathcal{A} = \{A, B, \ldots\} \subset \{s \approx t, P(t_1, \ldots, t_n) \mid P \in \mathcal{P}, s, t, t_i \in \mathcal{T}\} \qquad \text{atoms}$$
$$\mathcal{L} = \mathcal{A} \cup \{\neg A \mid A \in \mathcal{A}\} \qquad \text{literals}$$

Refutation-based theorem proving



Goal

A sound and refutation complete calculus.

Example

Apply resolution and factoring

$$\frac{A \vee \mathcal{C} \quad \neg B \vee \mathcal{D}}{(\mathcal{C} \vee \mathcal{D})\sigma}\ (\sigma)\ \mathsf{R} \qquad \frac{A \vee B \vee \mathcal{C}}{(A \vee \mathcal{C})\sigma}\ (\sigma)\ \mathsf{F}$$

$$\sigma = \mathrm{mgu}(A, B)$$

eventually on all clauses and literals until the empty clause is derived.

Observation

Search space grows too fast.

GOAL

A sound, refutation complete, and *effective* calculus.

1. *Reduce* search space
   - e.g. Ordered Resolution
   - ...selection functions for clauses and literals
2. *Reduce* redundancy
   - e.g. ignore clause $\mathcal{D}$, if $\mathcal{C}$ subsumes $\mathcal{D}$, i.e. $\mathcal{C}\tau \subseteq \mathcal{D}$.
   - ...depends on the calculus

Example (forward subsumption)

$$S = \{{}^{1:}\mathsf{P}(x, y), {}^{2:}\neg\mathsf{P}(\mathsf{a}, z)\} \cup \{{}^{3:}\mathsf{P}(\mathsf{a}, z')\}$$

t_1 subsumes t_3

$$\frac{\mathsf{P}(x, y) \quad \neg\mathsf{P}(\mathsf{a}, z)}{\square}\ \{x \mapsto \mathsf{a}, y \mapsto z\}$$

Resolution

$$S\bot = \{\mathsf{P}(\bot, \bot), \neg\mathsf{P}(\mathsf{a}, \bot), \mathsf{P}(\mathsf{a}, \bot)\}$$

InstGen/SMT

Goal

A sound, refutation complete, and *effective* calculus.

3. Quickly find
   - *variants* — variant removal
   - *instances* — backward subsumption
   - *generalizations* — forward subsumption
   - *unifiable terms* — resolution, etc.

   of a query term in a given set of terms.

Observation

Deduction rate drops quickly with linear search.

Position-Strings

Positions of a term

$$\mathcal{P}\mathsf{os}(t) = \begin{cases} \{\epsilon\} & \text{if } t = x \in \mathcal{V} \\ \{\epsilon\} \cup \{ip \mid 1 \leq i \leq n \wedge p \in \mathcal{P}\mathsf{os}(t_i)\} & \text{if } t = f(t_1, \ldots, t_n) \end{cases}$$

Traversals of $\mathsf{h}(\mathsf{f}(\mathsf{a}, y))$

$$\begin{array}{ccc} & \langle \epsilon, \mathsf{h} \rangle & \\ & \downarrow & \\ & \langle 1, \mathsf{f} \rangle & \\ \swarrow & & \searrow \\ \langle 11, \mathsf{a} \rangle & & \langle 12, \mathsf{y} \rangle \end{array}$$

$$\mathcal{P}\mathsf{os}(\mathsf{h}(\mathsf{f}(\mathsf{a}, \mathsf{y}))) = \{\epsilon, 1, 11, 12\}$$

$$\mathsf{h}(\mathsf{f}(\mathsf{a}, \mathsf{y}))|_{12} = y \qquad \langle 12, y \rangle$$

$\langle \epsilon, \mathsf{h} \rangle \langle 1, \mathsf{f} \rangle \langle 12, y \rangle$	root to leaf y	$(\mathsf{h}1\mathsf{f}2y)$
$\langle \epsilon, \mathsf{h} \rangle \langle 1, \mathsf{f} \rangle \langle 11, \mathsf{a} \rangle \langle 12, y \rangle$	pre-order	$(\mathsf{hfa}y)$

Normalization of variables

Forget variable names

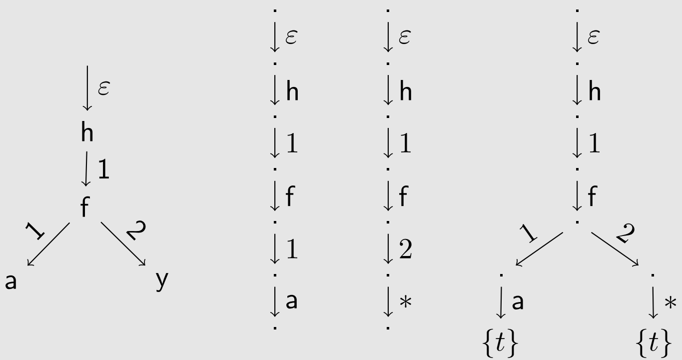
$$\begin{aligned} \text{path} : \mathsf{f}(x, y) &\mapsto \{\mathsf{f}1*, \mathsf{f}2*\} \\ \mathsf{f}(x, x) &\mapsto \{\mathsf{f}1*, \mathsf{f}2*\} \\ \text{pre-order} : \mathsf{f}(x, y) &\mapsto \mathsf{f}** \\ \mathsf{f}(x, x) &\mapsto \mathsf{f}** \end{aligned}$$

Enumerate variable names

$$\begin{aligned} \text{path} : \mathsf{f}(x, y) &\mapsto \{\mathsf{f}1*_1, \mathsf{f}2*_2\} \\ \mathsf{f}(x, x) &\mapsto \{\mathsf{f}1*_1, \mathsf{f}2*_1\} \\ \text{pre-order} : \mathsf{f}(x, y) &\mapsto \mathsf{f}*_1*_2 \\ \mathsf{f}(x, x) &\mapsto \mathsf{f}*_1*_1 \end{aligned}$$

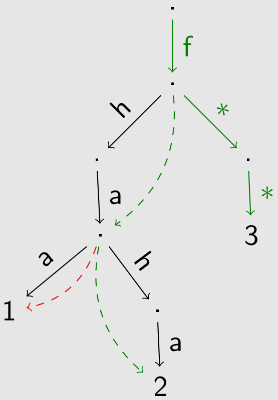
Path-Strings

The path-strings of $\mathsf{h}(\mathsf{f}(\mathsf{a}, y))$ are $\mathtt{h1f1a}$ and $\mathtt{h1f2}*$.



Pre-Order Strings

Non-linear terms

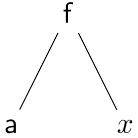


$\{ {}^{1:}\mathsf{f}(\mathsf{h}(\mathsf{a}), \mathsf{a}), {}^{2:}\mathsf{f}(\mathsf{h}(\mathsf{a}), \mathsf{h}(\mathsf{a}))\}, {}^{3:}\mathsf{f}(x, y)\}$

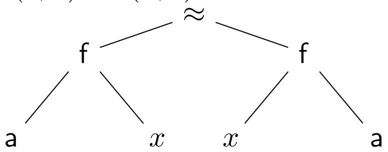
The terms $\mathsf{f}(x, y)$ and $\mathsf{f}(z, z)$ share the preorder-term $\mathsf{f}{*}{*}$.

But $\mathsf{f}(\mathsf{h}(\mathsf{a}), \mathsf{a})$ is not an instance of $\mathsf{f}(z, z)$.

$\mathsf{f}(\mathsf{a}, x)$

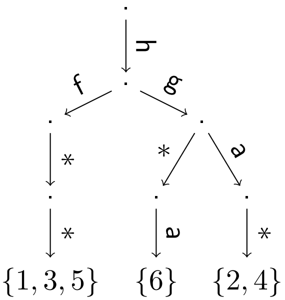


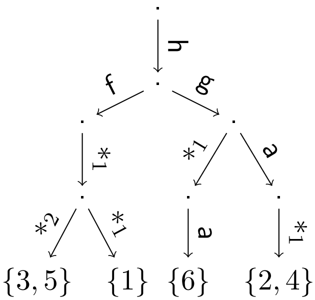
$\mathsf{f}(\mathsf{a}, x) \approx \mathsf{f}(x, \mathsf{a})$



Perfect filtering

$$\left\{ \begin{array}{lll} {}^{1:}\mathsf{h}(\mathsf{f}(x,x)), & {}^{2:}\mathsf{h}(\mathsf{g}(\mathsf{a},x)), & {}^{3:}\mathsf{h}(\mathsf{f}(y,z)), \\ {}^{4:}\mathsf{h}(\mathsf{g}(\mathsf{a},y)), & {}^{5:}\mathsf{h}(\mathsf{f}(y,x)), & {}^{6:}\mathsf{h}(\mathsf{g}(y,a)) \end{array} \right\}$$





$\mathsf{h}(\mathsf{f}(y,x)) \mapsto \mathsf{h.f.}{*}.{*}$

$\mathsf{h}(\mathsf{f}(y,x)) \mapsto' \mathsf{h.f.}{*_1}.{*_2}$