CS 511, Fall 2018, Handout 29

Unification

Assaf Kfoury

November 13, 2018

BACKGROUND

- The name “unification” and the first formal investigation of the notion is due to J.A. Robinson (1965).
- Robinson’s algorithm for first-order unification has exponential time-complexity in the worst-case.
- The Paterson-Wegman algorithm (1978) for first-order unification has linear time-complexity, but relatively complicated to implement.
- The Martelli-Montanari algorithm (1982) for first-order unification has a $\mathcal{O}(n \log n)$ time-complexity in the worst-case and is somewhat simpler to implement than the Paterson-Wegman algorithm.
- More information on first-order unification – the only kind we need in this course – can be found by browsing the Web. In particular, click here for an informative Wikipedia article.

  Problems of **unification** (and **matching**) are a rich and thriving area of computer science. Search the Web for: *semi-unification, acyclic semi-unification, second-order unification, bounded second-order unification, monadic second-order unification, context unification, stratified context unification,* and many other variants, each resulting from particular applications in computer science.

DEFINITIONS

- An ***instance*** of (first-order) unification is a finite set S of equations:

  $$S \triangleq \{s_1 \stackrel{?}{=} t_1, \ldots, s_n \stackrel{?}{=} t_n\}$$

  where $s_1, t_1, \ldots, s_n, t_n$ are first-order terms (over a given signature Σ).

- A ***substitution*** σ is always given as a mapping $\sigma : X \to \mathcal{T}$ where X is the set of all first-order variables and $\mathcal{T}$ is the set of all first-order terms.

  Such a substitution $\sigma : X \to \mathcal{T}$ is extended to $\sigma : \mathcal{T} \to \mathcal{T}$ in the usual way.

- A ***unifier*** or ***solution*** of S is a substitution σ such that $\sigma(s_i) = \sigma(t_i)$ for every $i = 1, \ldots, n$.

- *Sol*(S) is the set of all unifers or solutions of S. S is ***unifiable*** iff *Sol*$(S) \neq \varnothing$.

- A substitution σ is a ***most general unifier*** (***MGU***) of S if σ is a "least" element of *Sol*(S), *i.e.*, for every $\sigma' \in$ *Sol*(S) there is a substitution σ'' such that, for all variable x, it holds that $\sigma'(x) = \sigma''(\sigma(x))$ – more succintly written as $\sigma' = \sigma'' \circ \sigma$.

DEFINITIONS

- An ***instance*** of (first-order) unification is a finite set S of equations:

  $$S \triangleq \{s_1 \stackrel{?}{=} t_1, \ldots, s_n \stackrel{?}{=} t_n\}$$

  where $s_1, t_1, \ldots, s_n, t_n$ are first-order terms (over a given signature Σ).
- A ***substitution*** σ is always given as a mapping $\sigma : X \to \mathcal{T}$ where X is the set of all first-order variables and $\mathcal{T}$ is the set of all first-order terms.

  Such a substitution $\sigma : X \to \mathcal{T}$ is extended to $\sigma : \mathcal{T} \to \mathcal{T}$ in the usual way.
- A ***unifier*** or ***solution*** of S is a substitution σ such that $\sigma(s_i) = \sigma(t_i)$ for every $i = 1, \ldots, n$.
- *Sol*(S) is the set of all unifers or solutions of S. S is ***unifiable*** iff *Sol*$(S) \neq \varnothing$.
- A substitution σ is a ***most general unifier*** (***MGU***) of S if σ is a "least" element of *Sol*(S), *i.e.*, for every $\sigma' \in$ *Sol*(S) there is a substitution σ'' such that, for all variable x, it holds that $\sigma'(x) = \sigma''(\sigma(x))$ – more succintly written as $\sigma' = \sigma'' \circ \sigma$.
- Notational Conventions:
  1. We may write a substitution σ as the set of its *non-trivial bindings*, *i.e.*, $\sigma = \{\, x \mapsto \sigma(x) \mid \sigma(x) \neq x \,\}$.
  2. In particular, if we write $\sigma = \{\ \}$ (the empty set), then σ is the identity substitution.
  3. Whenever convenient and not ambiguous, we write "$\sigma\, t$" instead of "$\sigma(t)$".

AN ALGORITHM FOR FIRST-ORDER UNIFICATION

- We present an adaptation of the Martelli-Montanari algorithm, one of several available for first-order unification. (Its $\mathcal{O}(n \log n)$ time-complexity depends on some clever data structuring with dag's – not in this handout.)
- We can view unification as a rewrite system, the goal of which is to repeatedly transform a finite set of equations until the solution "stares you in the face".
- According to this view, unification can be carried using six transformation (or rewrite) rules (where the symbol "$\uplus$" denotes disjoint union):

AN ALGORITHM FOR FIRST-ORDER UNIFICATION

- We present an adaptation of the Martelli-Montanari algorithm, one of several available for first-order unification. (Its $\mathcal{O}(n \log n)$ time-complexity depends on some clever data structuring with dag's – not in this handout.)
- We can view unification as a rewrite system, the goal of which is to repeatedly transform a finite set of equations until the solution "stares you in the face".
- According to this view, unification can be carried using six transformation (or rewrite) rules (where the symbol "$\uplus$" denotes disjoint union):

[delete] $\{t \stackrel{?}{=} t\} \uplus S \implies S$

[decompose] $\{f(s_1,\ldots,s_m) \stackrel{?}{=} f(t_1,\ldots,t_m)\} \uplus S \implies \{s_1 \stackrel{?}{=} t_1,\ldots,s_m \stackrel{?}{=} t_m\} \cup S$

[conflict] $\{f(s_1,\ldots,s_m) \stackrel{?}{=} g(t_1,\ldots,t_n)\} \uplus S \implies$ FAIL
where $f \neq g$

[orient] $\{t \stackrel{?}{=} x\} \uplus S \implies \{x \stackrel{?}{=} t\} \cup S$
where $t \notin X$

[eliminate] $\{x \stackrel{?}{=} t\} \uplus S \implies \{x \stackrel{?}{=} t\} \cup \{x \mapsto t\}(S)$
where $x \notin$ *Var*(t) and $x \in$ *Var*(S)

[occurs check] $\{x \stackrel{?}{=} t\} \uplus S \implies$ FAIL
where $x \in$ *Var*(t) and $t \notin X$

(THIS PAGE INTENTIONALLY LEFT BLANK)