CS 576 lecture of 3/7/17

J. Meseguer

1. Motivation: Deciding QF satisfiability for $(\Sigma, \emptyset)$

Given a signature Σ, the theory $(\Sigma, \emptyset)$ is the theory of all Σ-algebras. In computer science this is confusingly called "the" theory of "uninterpreted function symbols". These notes will show that a decision procedure can be easily derived from standard results about Knuth-Bendix completion of a set of equations.

2. Knuth-Bendix Completion à la Huet

Knuth and Bendix wrote a seminal paper proposing an algorithm that, given an equational theory (Σ, E) as input and a chosen reduction order $>$ on terms will try to derive a rewrite theory $(\Sigma, \vec{Q})$ such that $E \vdash Q$ and $Q \vdash E$ (is an equivalent set of equations), and, furthermore, $\vec{Q}$ is confluent and $>$-terminating, i.e., for each $l \to r \in \vec{Q}$, $l > r$. Gerard Huet wrote a classical paper giving a simple description of the KB-algorithm and a proof of its correctness.

from CS 476

Recall that a reduction order (sometimes also called a simplification order) is a well-founded order $>$ on terms such that:

1. is <u>monotonic</u>: $(\forall f \in \Sigma_n)\ \forall u, v, t_1, .., t_n \in T_\Sigma(X)$,

$$u > v \Rightarrow f(t_1,.., \overset{i}{\check{u}},.., t_n) > f(t_1,.., \overset{i}{\check{v}},.., t_n)$$

2. <u>is closed under substitution</u>: $\forall \sigma \in [X \to T_\Sigma(X)]$, $\forall u, v \in T_\Sigma(X)$

$$u > v \Rightarrow u\sigma > v\sigma$$

For example, the <u>lexicographic path order</u>, and its generalization to the <u>reduction path order</u> (RPO) are reduction orders (see CS 476 for a detailed definition of the lexicographic path order).

Huet describes the KB algorithm as pseudocode. But it is more enlightened to describe it as an <u>inference system</u>, i.e., essentially as a set of rewrite rules transforming <u>configurations</u>, which are in general of the form:

marker for convergent rules

$$[E \mid R \mid m \mid j]$$

counter to generate new rules

current list of equations not yet oriented as rules

list of $>$-terminating and inter-reduced rules already computed, <u>indexed</u> by a natural number as label

where we always have $m < j$.

If we want to complete (Σ, E) into a convergent theory, we start with the configuration $[E, \emptyset, 0, 1]$.
The only final configurations are either $[R]$, a set of convergent rules equivalent to E, or [failure], if some equation could not be oriented as a rule wrt $>$.
Here is the two-rule inference system:

1. $[\, u = v \;\; E \mid R \mid m \mid j \,] \longrightarrow$

$\underline{\text{if}}\;\; u!_R = v!_R$

$\quad \underline{\text{then}}\; [\, E \mid R \mid m \mid j \,]$

$\quad \underline{\text{else}}\;\; \underline{\text{if}}\;\; u!_R \not> v!_R$ and $v!_R \not> u!_R$

$\qquad \underline{\text{then}}\; [\text{failure}]$

$\qquad \underline{\text{else}}\; [\, E\, E_1 \mid R_1 \mid m \mid s(j) \,]$

$\qquad\quad$ where $[j]\; l \to r$ is $[j]\, u!_R \to v!_R$ if $u!_R > v!_R$

$\qquad\qquad\qquad\qquad\qquad$ or $[j]\, v!_R \to u!_R$ otherwise

$\qquad\quad$ and $[E_1, R_1] =$ $\underline{\text{simplify}}$ R $\underline{\text{with}}$ $[j]\; l \to r$ $\underline{\text{ends}}$

$\qquad \underline{\text{fi}}$

$\underline{\text{fi}}$

and where $\underline{\text{simplify}}$ R $\underline{\text{with}}$ $[j]\; l \to r$ $\underline{\text{ends}}$ $=$

$=$ $\underline{\text{simplify-aux}}$ $(R :: R$ $\underline{\text{with}}$ $[j]\; l \to r :: [\text{nil}, \text{nil}]$

where $\underline{\text{simplify-aux}}$ is defined as follows:

$\underline{\text{simplify-aux}}$ $R :: nil$ $\underline{\text{with}}$ $[j]\ l \to r$ :: $[R, Q] = [R\ [j]\ l \to r, Q]$

$\underline{\text{simplify-aux}}$ $R :: [i]\ l' \to r'\ R'$ $\underline{\text{with}}$ $[j]\ l \to r$ :: $[R'', Q] =$

$\underline{\text{simplify-aux}}$ $R :: R'$ $\underline{\text{with}}$ $[j]\ l \to r$ :: $[R''', Q']$

where:

(i) if $l'!_{\{l \to r\}} \neq l'$ then $R''' = R''$ and $Q' = Q\ l'!_{\{l\to r\}} = r'$

(ii) otherwise, $R''' = R''\ [i]\ l' \to (r'!_{R\,l \to r})$ and $Q' = Q$.

2. $[\emptyset \mid R \mid m \mid j] \to$

$\underline{\text{if}}$ $next(R, m) = nil$

$\underline{\text{then}}$ $[R]$

$\underline{\text{else}}$ $[cps(l \to r, R_{\leq m}) \mid R \mid m' \mid j]$

where $[m']\ l \to r = next(R, m)$

fi

and where:

(i) $R_{\leq m}$ is the sublist of rules in R with index $\leq m$

(ii) $next(R, m)$ finds the smallest index m' with a rule $[m']\ l \to r$ in R, or yields nil otherwise

(iii) $cps(l \to r, R_{\leq}) = cps\text{-}aux(l \to r, R, R, nil)$

(iv) cps-aux is defined by the equations:

$$\text{cps-aux}(l \to r, R, nil, Q) = Q$$

$$\text{cps-aux}(l \to r, l' \to r' R, Q) = \text{cps-aux}(l \to r, R, Q\ \text{cps}_0(l \to r, l' \to r'; R))$$

(v) $\text{cps}_0(l \to r, l' \to r'; R)$ (the R-normalized <u>critical pairs</u> between $l \to r$ and $l' \to r'$)

$$= \langle r'\alpha !_{\cdot R} = l'[r]_p \alpha !_{\cdot R} \mid l' \overset{l \to r}{\underset{\alpha, p}{\rightsquigarrow}} l'[r]_p \alpha \wedge \alpha \in Unif(l, l'|_p) \wedge p \in pos_\Sigma(l') \rangle$$

$$\langle r\beta !_{\cdot R} = l[r']_q \beta !_{\cdot R} \mid l \overset{l' \to r'}{\underset{\beta, q}{\rightsquigarrow}} l[r']_q \beta \wedge \beta \in Unif(l', l|_q) \wedge q \in pos_\Sigma(l) \rangle$$

where $\langle exp \mid \varphi \rangle$ is a <u>list comprehension</u> notation expressing the list (as opposed to the set) obtained by computing the list elements exp that satisfy condition φ in some fixed order (in this case of considering positions p and computing unifiers α), and where $\langle exp \mid \varphi \rangle \langle exp' \mid \varphi' \rangle$ denotes the list concatenation of the two lists so obtained. Also, we assume that a "meta-equation" $(X = X) = nil$ has been added to the inference system, so that as soon as an equation $u = u$ appears in a list of equations it is immediately removed.

The key points about the correctness of the algorithm are the following *invariants*:

If $[E \mid \emptyset \mid 0 \mid 1] \xrightarrow{*} [E' \mid \vec{Q} \mid m \mid j]$ then:

(a) $E \vdash E' \cup Q$ and $E' \cup Q \vdash E$

(b) the rules $\vec{Q}_{\leq m}$ are convergent and inter-reduced

This then shows that when $[E \mid \emptyset \mid 0 \mid 1] \xrightarrow{*} [\emptyset \mid \vec{Q} \mid m \mid j]$ and $next(R, m) = nil$, we have $\vec{Q}$ convergent, and $E \vdash Q$ and $Q \vdash E$, as desired (see Huet's paper for more details).

Remark. Note that, although generally useful, it is not necessary to keep righthand sides of rules in R reduced (i.e., to keep rules "inter-reduced"). That is, we can instead define the function *simplify-aux* with a simpler last clause:

(ii)' otherwise $R''' = R'[i]\, l' \to r'$ and $Q' = Q$.

Nevertheless, the result of this new version of *simplify* will be a set R_1 of rules whose *lefthand sides* have been simplified by all other rules in R_1, given that this holds for the input R. That is, we get:

(b') the rules $\vec{Q}_{\leq m}$ are convergent with inter-reduced lefthand sides.

3. Ground Knuth-Bendix Completion

What happens when the input $[E \mid nil \mid 0 \mid 1]$ to the KB-completion algorithm is a list of ground equations? The two key observations are:

(i) there are reduction orders, for example RPO, that are total on ground terms, so we never fail!

(ii) $l \to r$ and $l' \to r'$ have a critical pair exactly when $l' \trianglelefteq l$, so that l becomes reducible by $l[l']_p \to l[r']_p$, or $l \trianglelefteq l'$, so that we get a reduction $l'[l]_q \to l'[r]_q$

But (i) and invariant (6) (resp. (6')) then means that when we reach

$$[E \mid nil \mid 0 \mid 1] \xrightarrow{*} [\emptyset \mid \vec{Q} \mid m \mid j]$$

since the rules $\vec{Q}$ are inter-reduced (resp. their lefthand sides are), $\vec{Q}$ has no critical pairs, and we can automatically terminate with a convergent $[\vec{Q}]$ such that $E \vdash Q$ and $Q \vdash E$.

4. The Plaisted - Sattler-Klein Ground KB Completion Algorithm

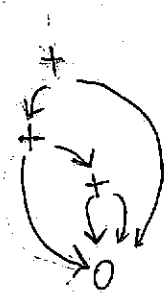
The algorithm by Plaisted and Sattler-Klein further refines ground KB-completion as follows:

(i) only lefthand sides are kept inter-reduced

(ii) the list of rules is sorted, so that rules with smaller righthand sides are given preference over others to process equations and rules

(iii) terms are represented as maximally shared dags

e.g., $(0 + (0 + 0)) + 0$ as



They then show that ground KB completion of a set of ground equations E of size n can be performed in $O(n^2)$ time.

5. Deciding QF satisfiability for (Σ, ϕ)

What does all this have to do with deciding satisfiability for the theory (Σ, ϕ) whose class of models is $\underline{Alg}_\Sigma$?

Everything! Note that, as usual, we can assume a QF φ in DNF as $\bigvee_i C_i$, with C_i a conjunction, so that it is enough to decide satisfiability of conjunctions. Then the main theorem is:

Theorem. Let $\varphi = \bigwedge_{i=1}^{n} u_i = v_i \wedge \bigwedge_{j=1}^{m} w_j \neq w_j'$ be a conjunction of equations and disequations. Then φ is satisfiable in (Σ, ϕ) iff

$$w_j !_{\vec{Q}} \neq w_j' !_{\vec{Q}} \qquad 1 \leq j \leq m$$

where $\vec{Q}$ is the KB-completion of the set of ground $\Sigma(Y)$-equations $\{u_i = v_i \mid 1 \leq i \leq n\}$, where $Y = vars(\varphi)$, ~~and~~ we treat the variables Y as extra constants.

Proof

($\Leftarrow$) consider the Σ-algebra $\mathcal{C}_{\Sigma(Y),\vec{Q}}|_{\Sigma}$. It obviosly satisfies φ with the assignment $\eta_Y : Y \ni y \mapsto y!_{\vec{Q}} \in C_{\Sigma(Y),\vec{Q}}$ that makes it into a $\Sigma(Y)$-algebra.

($\Leftarrow$) Suppose an assignment $a : Y \to A$ for a Σ-algebra $\mathbb{A} = (A, \_{}_{\mathbb{A}})$ satisfies φ. This exactly means that:

(i) the $\Sigma(Y)$-algebra $(\mathbb{A}, a)$ satisfies the ground equations $\{u_i = v_i \mid 1 \leq i \leq n\}$

(ii) it also satisfies the ground disequations $\{w_j \neq w'_j \mid 1 \leq j \leq m\}$

But (i), and the isomorphism $\mathcal{T}_{\Sigma(Y)/\{u_i = v_i \mid 1 \leq i \leq n\}} \cong \mathcal{C}_{\Sigma(Y),Q}$ means that there is a unique $\Sigma(Y)$-homomorphism

$$\_{}_{(\mathbb{A},a)} : \mathcal{C}_{\Sigma(Y),Q} \longrightarrow (\mathbb{A}, a)$$

and this forces $w_j!_{\vec{Q}} \neq w'_j!_{\vec{Q}}$, $1 \leq j \leq m$, since otherwise, if we had $w_k!_{\vec{Q}} = w'_k!_{\vec{Q}}$ for some k, $1 \leq k \leq m$, we would have $w_k!_{Q\,(\mathbb{A},a)} = w_k a \neq w'_k a = w'_k!_{Q\,(\mathbb{A},a)}$ contradicting $w_k!_{\vec{Q}} = w'_k!_{\vec{Q}}$. $\square$