A system for solving equational constraints Examples and experimental results

Nicolas Peltier
Laboratory LEIBNIZ-IMAG
46, Avenue Félix Viallet 38031 Grenoble Cedex FRANCE
Nicolas.Peltier@imag.fr
Phone: (33) (0)4 76 57 48 05

December 1, 1997

1 A first example

We give an example of input file for the system $\text{ECS}_{\text{ATINF}}$. Let $\mathcal{S} = \{nat, list\}$ be a set of sort symbols. Let Σ be a signature containing two constant symbols 0 and *nil* of sort *nat* and *list* respectively and two function symbols *succ* and *cons*, of profile $nat \rightarrow nat$ and $nat \times list \rightarrow list$. Let $\mathcal{F}$ be the following equational formula

$$\mathcal{F} = (\forall y.x \neq succ(y) \wedge z = cons(x,u) \wedge \forall v,w.u \neq cons(v,cons(w,cons(x,nil))))$$

The corresponding input file is the following:

```
% Definition of function symbols.
declare(nat(zero)).
declare(nat(succ(nat))).
declare(list(cons(nat,list))).
declare(list(nil)).

 signature = { succ, zero, cons, nil }.

formula(f(X,U,Z)).
((all Y (X != succ(Y))) & (Z = cons(X,U)) &
(all V (all W (U != cons(V,cons(W,cons(X,nil))))))).

solve(f).
```

ECS$_{\text{ATINF}}$ gives the following result:

```
-------------  ECS-ATINF -------------
--- Solving Equational Constraints ---
---------------------------------------

Please report bugs or problems to Nicolas Peltier
(e-mail: Nicolas.Peltier@imag.fr)

Loading file tt

Defining new sort symbol nat.
Defining new function symbol zero :  -> nat

Defining new function symbol succ : nat -> nat

Defining new sort symbol list.
Defining new function symbol cons : nat , list -> list

Defining new function symbol nil :  -> list


Expand formula
((all C -(A = succ(C))) & (D = cons(A,B))
& (all E (all F -(B = cons(E,cons(F,cons(A,nil)))))))
Renumber formula
((all C -(A = succ(C))) & (D = cons(A,B))
& (all E (all F -(B = cons(E,cons(F,cons(A,nil)))))))
Type verification for formula
((all C -(A = succ(C))) & (D = cons(A,B))
& (all E (all F -(B = cons(E,cons(F,cons(A,nil)))))))

Defining new formula f

---- Solving equational formula:
((all C -(A = succ(C))) & (D = cons(A,B))
& (all E (all F -(B = cons(E,cons(F,cons(A,nil)))))))

Transformation into negation normal form
((all C -(A = succ(C))) & (D = cons(A,B))
& (all E (all F -(B = cons(E,cons(F,cons(A,nil)))))))
Eliminating parameters for equational problem:
(all C -(A = succ(C)))
```

```
Elimination of parameters
-(A = succ(C))
Formula is:
-(A = succ(C))
Applying the explosion rule on A
Formula is:
((-(C = M2) & (A = succ(M2))) | (A = zero))
Applying the universality of parameters rule on -(C = M2)
Formula is:
(A = zero)
Eliminating parameters for equational problem:
(all E (all F -(B = cons(E,cons(F,cons(A,nil))))))
Elimination of parameters
-(B = cons(E,cons(F,cons(A,nil))))
Formula is:
-(B = cons(E,cons(F,cons(A,nil))))
Applying the explosion rule on B
Formula is:
(((-(E = M2) | -(C = cons(F,cons(A,nil))))
& (B = cons(M2,C))) | (B = nil))
Applying the universality of parameters rule on -(E = M2)
Formula is:
((-(C = cons(F,cons(A,nil))) & (B = cons(M2,C)))
| (B = nil))
Applying the explosion rule on C
Formula is:
(((((-(F = O2) | -(N2 = cons(A,nil)))
& (C = cons(O2,N2))) | (C = nil)) & (B = cons(M2,C))) | (B = nil))
Applying the universality of parameters rule on -(F = O2)
Formula is:
((((-(N2 = cons(A,nil)) & (C = cons(O2,N2))) | (C = nil))
& (B = cons(M2,C))) | (B = nil))
Solving equational problem:
((exists N2 (exists C (exists O2 (exists M2 ((((-(N2 = cons(A,nil))
& (C = cons(O2,N2))) | (C = nil)) & (B = cons(M2,C))) | (B = nil))))))
& (D = cons(A,B)) & (A = zero))
Elimination of parameters
(((((-(N2 = cons(A,nil)) & (C = cons(O2,N2))) | (C = nil))
& (B = cons(M2,C))) | (B = nil)) & (D = cons(A,B)) & (A = zero))
Formula is:
(((((-(N2 = cons(zero,nil)) & (C = cons(O2,N2))) | (C = nil))
& (B = cons(M2,C))) | (B = nil)) & (D = cons(zero,B)) & (A = zero))
Transformation into definition with constraints
```

```
(((((-(N2 = cons(zero,nil)) & (C = cons(O2,N2))) | (C = nil))
& (B = cons(M2,C))) | (B = nil)) & (D = cons(zero,B)) & (A = zero))
Formula is:
(((((-(N2 = cons(zero,nil)) & (C = cons(O2,N2))) | (C = nil))
& (B = cons(M2,C))) | (B = nil)) & (D = cons(zero,B)) & (A = zero))
Formula is:
((((C = cons(O2,N2)) & -(N2 = cons(zero,nil))
& (B = cons(M2,cons(O2,N2)))) | ((C = nil)
& (B = cons(M2,nil))) | (B = nil))
& (D = cons(zero,B)) & (A = zero))
Formula is:
(((B = nil) & (D = cons(zero,nil)) & (A = zero)) |
((B = cons(M2,nil)) & (C = nil) & (D = cons(zero,cons(M2,nil))) & (A = zero)) |
((B = cons(M2,cons(O2,N2))) & -(N2 = cons(zero,nil))
& (C = cons(O2,N2)) & (D = cons(zero,cons(M2,cons(O2,N2)))) & (A = zero)))
Formula is:
(((B = nil) & (D = cons(zero,nil)) & (A = zero)) |
((B = cons(M2,nil)) & (D = cons(zero,cons(M2,nil))) & (A = zero))
| ((B = cons(M2,cons(O2,N2))) & -(N2 = cons(zero,nil))
& (D = cons(zero,cons(M2,cons(O2,N2)))) & (A = zero)))

f = (exists M2 (exists O2 (exists N2 (((B = nil)
& (D = cons(zero,nil)) & (A = zero))
| ((B = cons(M2,nil)) & (D = cons(zero,cons(M2,nil))) & (A = zero))
| ((B = cons(M2,cons(O2,N2))) & -(N2 = cons(zero,nil))
& (D = cons(zero,cons(M2,cons(O2,N2)))) & (A = zero))))))

Run Time:     0.03

--- BEGIN INTERACTION ---

ECS-ATINF >
---------- Bye ----------
```

ECS$_{\text{ATINF}}$ can also be used to verify that a substitution is a solution of a given formula. For example, the following commands check that the substitution $\{x \to 0, z \to cons(0, nil), u = 0\}$ is a solution of $\mathcal{F}$:

```
formula(q).
f(0,cons(0,nil),0).

solve(f).
```

```
q = TRUE

Run Time:     0.00
```

2 Using $\text{Ecs}_{\text{Atinf}}$ for model verification

A Herbrand interpretation $\mathcal{I}$ is an eq-interpretation [1] iff for each predicate symbol P of arity n, there exists an equational formula $\mathcal{F}_P$ containing n free variables $x_1, \ldots, x_n$ such that, for any terms $t_1, \ldots, t_n$, $\mathcal{I} \models P(t_1, \ldots, t_n)$ iff the substitution $\{x_1 \rightarrow t_1, \ldots, x_n \rightarrow t_n\}$ is a solution of $\mathcal{F}_P$.

Remark: atomic interpretations [2] are particular cases of eq-interpretation.

ECS$_{\text{ATINF}}$ can be used to decide whether an eq-interpretation $\mathcal{I}$ is a model of a given first-order formula $\mathcal{F}$. We give below an example, adapted from [3].

```
% The following example is taken from (Golbfarb, 1983)
% (Journal of Symbol Logic, vol 4, number 4, pages 1237-1252).

% We first define a Herbrand interpretation
% by defining the interpretation of each predicate symbol.

% Then we define a first-order formula and we use
% the constraint solving algorithm for finding the
% truth value of the formula in the given
% interpretation.

% Definition of constructors.

clear(verbose).
declare(nat(succ(nat))).
declare(nat(zero)).
declare(pair(cons(nat,nat))).

signature = { succ, zero, cons }.

% Interpretation of predicate symbols.

% z(X) means X = 0.

formula(z(X)).
(X = zero).

% s(X,Y) means that X is the successor of Y.
formula(s(X,Y)).
```

```
(X = succ(Y)).

% p1(X,Y) means that Y is the first component of X
formula(p1(X,Y)).
(exists V (X = cons(Y,V))).

% p2(X,Y) means that Y is the second component of X
formula(p2(X,Y)).
(exists V (X = cons(V,Y))).

% r1(X,Y) means that the first component of Y is the
% successor of the second component of X.
formula(r1(X,Y)).
(exists U (exists V (exists W ((X = cons(U,V)) & (Y = cons(succ(V),W)))))).

% r2(X,Y) means that the second component of Y is the
% successor of the second component of X.
formula(r2(X,Y)).
(exists U (exists V (exists W ((X = cons(U,V)) & (Y = cons(W,succ(V))))))).

% n(X,Y) means that the first component of Y is the
% successor of the first component of X and that the
% second components of X and Y are equal.
formula(n(X,Y)).
(exists U (exists V ((X = cons(U,V)) & (Y = cons(succ(U),V))))).

% q(X,Y) means that Y is the successor of the second component of X.
formula(q(X,Y)).
(exists U (exists V ((X = cons(U,V)) & (Y = succ(V))))).

formula(f).

((all X (all Y ((z(X) & z(Y)) -> (X = Y))))
& (all X (exists Z (n(X,Z) -> ((all Y (q(X,Y) -> q(Z,Y)))
& (all Y (r1(X,Y) -> r1(Z,Y))) & (all Y (r2(X,Y) -> r2(Z,Y)))))))
& (all X (all Y (n(X,Y)
-> (exists W (exists U (p1(X,W) & s(U,W) & p1(Y,U)))))))
& (all X (all Y (n(X,Y)
-> (exists Z (p2(X,Z) & p2(Y,Z))))))
& (all X (all Y (q(X,Y)
-> (exists Z (p1(X,Z) & (s(Y,Z) -> p2(X,Z)))))))
& (all X (all Y (r1(X,Y)
-> (exists Z (exists W (p1(X,Z) & s(W,Z) & (p1(Y,W) -> p2(X,Z))))))))
& (all X (all Y (r2(X,Y)
-> (exists Z (exists W (p1(X,Z) & s(W,Z) & (p2(Y,W) -> p2(X,Z))))))))
```

```
& (all X -s(zero,X))
& (all X (all Y ((-z(X) & (X != Y))
-> (exists Z (s(X,Z) & -s(Y,Z))))))
& (all X (exists Y s(Y,X)))
& z(zero)
& (all X (all Y ((p1(X,zero) & p1(X,Y)) -> (Y = zero))))
& (all X (all Y (s(X,Y) -> (exists U (q(U,X) & p2(U,Y) & p1(U,zero))))))
& (all X (all Y ((p1(X,Y) & -z(Y))
-> (exists U (exists W (r1(U,X) & p2(U,W) & p1(U,zero) & s(Y,W)))))))
& (all X (all Y ( (p1(X,Y) & -z(Y))
-> (exists U (exists W (r1(U,X) & p2(U,W) & p1(U,zero) & s(Y,W)))))))
& (all X (all Y ( (p2(X,Y) & -z(Y))
-> (exists U (exists W ( r2(U,X) & p2(U,W) & p1(U,zero) & s(Y,W)))))))
& (all X (all Y ((p2(X,zero) & p2(X,Y))
-> (Y = zero))))).

solve(f).
quit.
```

We obtain the following result:

```
f = TRUE

Run Time: 0.09
```

The formula is valid, which proves that the initial interpretation is a model of the formula f (see [4] for details).

Some experimental results

Figure 1 and 2 present some equational formulae given by H. Comon (private communication) and the solved forms obtained with $\text{ECS}_{\text{ATINF}}$. We also give the running time (on a SUN SPARC 5).

References

[1] R. Caferra and N. Zabel. A method for simultaneous search for refutations and models by equational constraint solving. *Journal of Symbolic Computation*, 13:613–641, 1992.

[2] C. Fermüller and A. Leitsch. Hyperresolution and automated model building. *Journal of Logic and Computation*, 6(2):173–203, 1996.

[3] W. D. Goldfarb. The unsolvability of the Gödel class with identity. *Journal of Symbolic Logic*, 49(4):1237–1252, 1984.

Problème	Définition
1	$0 = 0 \vee (x = y \wedge x \neq z)$
2	$0 \neq 0 \vee (x = y \wedge x \neq z)$
3	$\forall y.s(x) \neq s(y) \vee x = y$
4	$f(x_1, x_2) = f(x_2, x_1)$ $\vee g(x_1, x_2) = g(x_2, x_1) \vee x_1 \neq s(x_3)$
5	$f(y, y) = f(x(z), z) \vee f(x, y) = f(y, s(x)) \vee z \neq s(y)$
6	$x = y \wedge (z \neq s(x) \vee x = s(z))$
7	$\forall y.x \neq f(f(y, s(y)), s(s(y)))$
8	$x = y \wedge x = z$
9	$x = s(s(z)) \wedge x = s(y) \wedge y = s(s(x)) \wedge z = s(s(y))$
10	$\forall y.(\exists x, z.(x = s(z) \vee x \neq z) \wedge x \neq s(s(z)))$
11	$\forall y.(y = s(x) \wedge z = s(x)) \vee (\exists z.z \neq s(x) \wedge z \neq x)$
12	$\forall y.((x \neq s(s(y)) \vee x \neq s(0)) \wedge z = s(x))$ $\vee(\exists z.z \neq s(x) \wedge z \neq x)$
13	$\forall y_1, y_2, y_3, y_4.$ $(y_1 = x_1 \vee f(x_1, x_4, x_4) = f(x_1, x_2, s(x_3)))$ $\wedge(f(y_1, y_2, s(y_2)) \neq f(s(y_2), x_1, x_2))$ $\wedge x_1 \neq f(y_1, y_2, y_3)$
14	$\forall y.s(x) = s(z) \wedge (x = z \vee x \neq s(y))$
15	$\forall y_1, y_2, y_3, y_4.(y_2 \neq f(y_1, y_1) \vee y_3 \neq f(y_2, y_2)$ $\vee y_4 \neq f(y_3, y_3) \vee x \neq f(y_1, y_1)$
16	$(\exists x.\forall y.\exists z.(x \neq 0 \wedge x \neq f(y, z)))$
17	$\forall y.x \neq y \vee x \neq s(y)$

Figure 1: Examples of equational formulae

Formula	Solved Form	Times (in ms)
1	$\top$	0.65
2	$y \neq z \wedge x = y$	0.85
3	$\top$	0.68
4	$x_1 \neq s(x_3) \vee x_2 \neq s(x_3)$	1.17
5	$z \neq s(y)$	0.93
6	$x = y \wedge x = z$	0.91
7	$(\exists u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9, u_{10}$ $(x = f(u_1, s(f(u_2, u_3))))$ $\vee(x = f(u_1, s(0)))$ $\vee(x = f(u_1, 0))$ $\vee(x = f(u_1, f(u_4, u_5)))$ $\vee(u_6 \neq s(s(u_7)) \wedge (x = f(f(u_7, u_8), u_6)))$ $\vee(u_6 \neq s(s(u_7)) \wedge (x = f(0, u_6)))$ $\vee(u_6 \neq s(s(u_7)) \wedge (x = f(s(u_9), u_6)))$ $\vee(u_8 \neq s(u_7) \wedge (x = f(u_1, s(f(u_2, u_3)))))$ $\vee(u_8 \neq s(u_7) \wedge (x = f(u_1, s(0))))$ $\vee(u_8 \neq s(u_7) \wedge (x = f(u_1, 0)))$ $\vee(u_8 \neq s(u_7) \wedge (x = f(u_1, f(u_4, u_5))))$ $\vee(u_8 \neq s(u_7) \wedge (x = f(f(u_7, u_8), u_6)))$ $\vee(u_8 \neq s(u_7) \wedge (x = f(0, u_6)))$ $\vee(u_8 \neq s(u_7) \wedge (x = f(s(u_9), u_6)))$ $\vee(x = f(0, u_6))$ $\vee(x = f(s(u_9), u_6))$ $\vee(x = s(u_{10}))$ $\vee(x = 0))$	16.69
8	$z = y \wedge x = z$	0.65
9	$\bot$	0.95
10	$\top$	1.36
11	$\exists z.(z \neq x \wedge z \neq s(x))$	1.52
12	$\exists u.((u \neq x) \wedge (u \neq s(x)) \vee (z = s(x)))$	1.75
13	$\exists u.((x = 0) \wedge (x_4 = x_2)$ $\wedge(x_2 = s(x_3))) \wedge (x_1 = s(u) \wedge u \neq x_3)$ $\wedge(x_4 = x_2) \wedge (x_2 = s(x_3))$	8.32
14	$x = z$	0.97
15	$\exists u_1, u_2.(u_1 \neq u_2 \wedge x = f(u_1, u_2))$ $\vee x = 0$	2.18
16	$\exists x.x \neq 0$	1.14
17	$\top$	0.71

Figure 2: Examples of equational formulae (solved forms)

[4] N. Peltier. *Nouvelles Techniques pour la Construction de Modèles finis ou infinis en Déduction Automatique*. PhD thesis, Institut National Polytechnique de Grenoble, 1997.