Konwersja logiki temporalnej

Marcel Lekston
Filip Prasałek

Jako przykład konwersji logiki temporalnej rozpatrzyliśmy problem "kierowcy w samochodzie". Wiedząc, że jeżeli kierowca znajduje się w środku, to samochód jest w ruchu, należy wywnioskować, że jeżeli człowiek nie jest w środku, to samochód nigdy nie ruszy. Teza, sformułowana w postaci logiki temporalnej, wygląda następująco:

$$\neg r \wedge G_f$$

Po konwersji na First Order Logic stosując następujące reguły:

$$\begin{array}{lll} p & \rightsquigarrow & p(t) \\ \Diamond p & \rightsquigarrow & \exists t'.\ (t' \geq t) \wedge p(t') \\ \Box p & \rightsquigarrow & \forall t'.\ (t' \geq t) \Rightarrow p(t') \end{array}$$

gdzie

$\Diamond \rightsquigarrow F$ sometime in the Future

$\Box \rightsquigarrow G$ Globally in the future

Otrzymujemy następującą formułę:

$$\neg r(t) \wedge (\forall_d (d \geq t)) \implies f(d)$$

Po odpowiednim przekształceniu do formatu provera, otrzymujemy następujący input:

```
begin_problem(Driver).
list_of_descriptions. name({*Driver and car*}).
author({*FP & ML*}).
status(unsatisfiable).
description({* Car will never move unless driver is inside *}).
end_of_list.

list_of_symbols. functions[(t1,0),(t2,0)].
predicates[(Autostoi,1),(Kierowcawaucie,1),(Greaterequal,2)].
end_of_list.

 list_of_formulae(axioms).
formula(Greaterequal(t2,t1)).
formula(not(Kierowcawaucie(t1))).
formula(forall([d,t],implies(and(Greaterequal(d,t),not(Kierowcawaucie(t))),Autostoi
(d)))).
end_of_list.

list_of_formulae(conjectures).
formula(Autostoi(t2)).
end_of_list.
end_problem.
```

W wyniku tego otrzymujemy:

```
--------------------------SPASS-START-----------------------------
Input Problem:
1[0:Inp] || Autostoi(t2)* -> .
2[0:Inp] ||  -> Greaterequal(t2,t1)*.
3[0:Inp] || Kierowcawaucie(t1)* -> .
4[0:Inp] || Greaterequal(u,v)* -> Autostoi(u) Kierowcawaucie(v).
 This is a first-order Non-Horn problem without equality.
 This is a problem that has, if any, a finite domain model.
 There are no function symbols.
 The conjecture is ground.
 Axiom clauses: 3 Conjecture clauses: 1
 Inferences: IORe=1 IOFc=1
 Reductions: RFMRR=1 RBMRR=1 RObv=1 RUnC=1 RTaut=1 RFSub=1 RBSub=1 RCon=1
 Extras    : Input Saturation, Always Selection, Full Splitting, Full Reduction,
Ratio: 5, FuncWeight: 1, VarWeight: 1
 Precedence: Autostoi > Kierowcawaucie > Greaterequal > t1 > t2
 Ordering  : KBO
Processed Problem:

Worked Off Clauses:

Usable Clauses:
2[0:Inp] ||  -> Greaterequal(t2,t1)*.
```

```
1[0:Inp] || Autostoi(t2)* -> .
3[0:Inp] || Kierowcawaucie(t1)* -> .
5[0:Res:4.1,1.0] || Greaterequal(t2,u)* -> Kierowcawaucie(u).
4[0:Inp] || Greaterequal(u,v)* -> Kierowcawaucie(v) Autostoi(u).
        Given clause: 2[0:Inp] ||  -> Greaterequal(t2,t1)*.
        Given clause: 1[0:Inp] || Autostoi(t2)*+ -> .
        Given clause: 3[0:Inp] || Kierowcawaucie(t1)*+ -> .
        Given clause: 5[0:Res:4.1,1.0] || Greaterequal(t2,u)*+ ->
Kierowcawaucie(u).
SPASS V 3.9
SPASS beiseite: Proof found.
Problem: Read from stdin.
SPASS derived 2 clauses, backtracked 0 clauses, performed 0 splits and kept 6
clauses.
SPASS allocated 72519 KBytes.
SPASS spent	0:00:02.37 on the problem.
		0:00:02.31 for the input.
		0:00:00.01 for the FLOTTER CNF translation.
		0:00:00.00 for inferences.
		0:00:00.00 for the backtracking.
		0:00:00.00 for the reduction.

--------------------------SPASS-STOP------------------------------
```