Artificial Intelligence 1 – WS 2020/2021
Assignment 6: Logic
– Given Dec 17, Due Jan 10 –

Problem 6.1 (PL Concepts) 30 pt

Which of the following statements are true? In each case, give an informal argument why it is true or a counter-example.

1. Every satisfiable formula is valid.
2. Every valid formula is satisfiable.
3. If A is satisfiable, then $\neg A$ is unsatisfiable.
4. If $A \models B$, then $A \wedge C \models B \wedge C$.
5. Every admissible inference rule is derivable.
6. If $\vdash$ is sound for $\models$ and $\{A, B\} \vdash C$, then C is satisfiable if A and B are.

Problem 6.2 (Equivalence of CSP and SAT) 30 pt

We consider

- CSPs (V, D, C) with finite domains as before
- SAT problems (V, A) where V is a set of propositional variables and A is a propositional formula over V.

We will show that these problem classes are equivalent by reducing their instances to each other.

1. Given a SAT instance $P = (V, A)$, define a CSP instance $P' = (V', D', C')$ and two bijections

   - f mapping satisfying assignments of P to solutions of P'
   - f' the inverse of f

   We already know that binary CSPs are equivalent to higher-order CSPs. Therefore, it is sufficient to give a higher-order CSP.

2. Given a CSP instance (V, D, C), define a SAT instance (V', A') and bijections as above

Problem 6.3 (Calculi Comparison) 60 pt

Prove (or disprove) the validity of the following formulae in i) Natural Deduction ii) Tableau and iii) Resolution.

1. $(P \wedge Q) \Rightarrow (P \vee Q)$ (to be done in the tutorial, not part of grading)
2. $((A \vee B) \wedge (A \Rightarrow C) \wedge (B \Rightarrow C)) \Rightarrow C$
3. $((P \Rightarrow Q) \Rightarrow P) \Rightarrow P$

Problem 6.4 (Substitution in Prolog) 30 pt

Implement substitution for first-order logic in Prolog by completing the stub below.

You do not have to avoid variable capture. Explain the problem of variable capture using the three test inputs as examples.

```
% We use a simple fragment of first−order logic
% truth for truth
% conj(F,G) for conjunction
% univ(x,F) for universal quantification
% applyF(p,L) for applications of predicate p (a string) to list L of terms
% vr(x) for variables where x is a string
% applyT(f,L) for applications of function f (a string) to list L of terms

% We represent substitutions as list S=[(x,t),...] where x is a variable and t a term

% lookup(S,X,T) hold if S contains (X,T)

% subsF(F,S,F2) holds if formula F2 arises from F via substitution S

% subsT(T,S,T2) holds if term T2 arises from T via substitution S

% subsTs(Ts,S,T2) holds list of terms Ts2 arises from Ts via substitution S

% test inputs:
% subsF(univ("x",applyF("p",[vr("x")])),[("x",vr("y"))],F)
% subsF(univ("y",applyF("p",[vr("x")])),[("x",vr("y"))],F)
% subsF(univ("y",applyF("p",[vr("x")])),[("x",vr("z"))],F)
```

relations among inds.

constants, variables, functions and predicates