

5

Language

- A unary function *color*, where *color*(*x*) is the color associated to the node *x*
- A unary predicate *node*, where *node*(*x*) means that *x* is a node
- A binary predicate *edge*, where *edge*(*x*, *y*) means that *x* is connected to *y*

Axioms

1. $\forall x \forall y. (\text{edge}(x, y) \rightarrow (\text{color}(x) \neq \text{color}(y)))$
"Two connected node are not equally colored."
2. $\forall x \forall x_1 \dots \forall x_{k+1}. \left(\bigwedge_{h=1}^{k+1} \text{edge}(x, x_h) \rightarrow \bigvee_{i,j=1, j \neq i}^{k+1} x_i = x_j \right)$
"A node does not have more than *k* connected nodes."

6

```
tree_count(nil,0).
tree_count(t(_,nil, nil),0).
tree_count(t(_,L,R),N):-
    tree_count(L,N1), tree_count(R,N2), N is N1+N2+1.
```

Please note that the first clause is needed, otherwise the program does not compute the correct number. However, I have considered correct also the solutions without such a first clause.

Also note that this program gives the correct solution only with the first computed answer substitution, then if one asks for more solutions one gets incorrect answers. To avoid this one should use the cut (!) in the first and the second clause.

```
7  q(W,b,Y) -> (using rule 1 and unifier {X/b}, W/a) }
    q(b,b,Y) -> (using rule 3 and unifier Z'/Y)
    p(Y),r(Y) -> (using rule 4 and unifier Y/f(V))
    r(f(V)) -> (using rule 5 and unifier V/a)
    [] (empty clause)
```

The answer is ~~W/b,X/Y,a~~ {W/a, Y/f(a)}

8. Generate and test means that the program first generate the values for the variables and then check whether the constraint are satisfied (if not, backtracking occurs).

Constraint and generate means that the program first impose the constraints, which are used to limit the search space, and afterwards the values for instantiating the variables are generated.