

```

1.     private void ResolveBacksubstitutions() {
2.         var constantFolder = new ConstantFoldingVisitor();
3.
4.         // The "ToArrays" are necessary because we modify _closedVariables below. Without them,
5.         // we get InvalidOperationExceptions telling us that the "Collection was modified."
6.         var varsWithValue = _closedVariables.Values.OfType<VariableWithValue>().ToArray();
7.         var varsWithBacksubstitutions = _closedVariables.Values.OfType<VariableWithBacksubstitution>().ToArray();
8.
9.         while (varsWithBacksubstitutions.Any()) {
10.            if (!varsWithValue.Any()) {
11.                Debug.WriteLine("---- No complete backsubstitution possible ----");
12.                foreach (var vb in varsWithBacksubstitutions) {
13.                    Debug.WriteLine(vb);
14.                }
15.                throw new InvalidOperationException("No solution found - " +
16.                    "there are more variables to substitute, but still open substitutions");
17.            }
18.
19.            var newVarsWithValue = new List<VariableWithValue>();
20.            var newVarsWithBacksubstitutions = new List<VariableWithBacksubstitution>();
21.            foreach (var varWithValue in varsWithValue) {
22.                // We substitute each variable into all open backsubstitutions.
23.                var rewriter = new RewritingVisitor(new Dictionary<AbstractExpr, AbstractExpr>
24.                    { { varWithValue.Variable,
25.                      new Constant(varWithValue.Value)
26.                    } });
27.                foreach (var varWithBacksub in varsWithBacksubstitutions) {
28.                    AbstractExpr rewritten = varWithBacksub.Expr
29.                        .Accept(rewriter, Ig.nore)
30.                        .Accept(constantFolder, Ig.nore);
31.                    // If the result, after constant folding, is a constant, we have found a new solution value.
32.                    // Otherwise, we still have a - maybe smaller - backsubstitution for this variable.
33.                    if (rewritten is Constant) {
34.                        var result = new VariableWithValue(varWithBacksub.Variable, ((Constant)rewritten).Value);
35.                        newVarsWithValue.Add(result);
36.                        _closedVariables[varWithBacksub.Variable] = result;
37.                    } else {
38.                        var result = new VariableWithBacksubstitution(varWithBacksub.Variable, rewritten);
39.                        newVarsWithBacksubstitutions.Add(result);
40.                    }
41.                }
42.            }
43.            varsWithValue = newVarsWithValue.ToArray();
44.            varsWithBacksubstitutions = newVarsWithBacksubstitutions.ToArray();
45.        }
46.    }

```