

Solves informaticus Cryptarithm puzzle using SAT solver.



In[*]:=

;



In[*]:=

;

In[*]:= imgs = {};

In[*]:= imgToVar = AssociationThread[imgs, v /@ Range@Length@imgs];
varToImg = AssociationThread[Values@imgToVar, Keys@imgToVar];

```

In[*]:= eqs = Hold@{
   +  == ,
   +  ==  * 10 + ,
   +  == 10 *  + ,
   +  == ,
   +  == ,
   + 10 *  +  ==  * 10 + ,
   +  == ,
   +  ==  * 10 + ,
   +  * 10 +  ==  * 10 + ,
   +  == ,
   +  ==  * 10 + ,
   +  * 10 +  ==  * 10 + 
};

```

```

In[*]:= eqs = eqs /. imgToVar // ReleaseHold

```

```

Out[*]=

```

```

{v[1] + v[2] == v[3], v[7] + v[8] == 10 v[1] + v[10], v[5] + v[9] == 10 v[1] + v[3],
 v[1] + v[8] == v[9], v[2] + v[7] == v[5], 10 v[1] + v[3] + v[10] == 10 v[1] + v[3],
 v[1] + v[2] == v[3], v[4] + v[5] == 10 v[1] + v[10], 10 v[1] + v[6] + v[7] == 10 v[1] + v[3],
 v[1] + v[4] == v[6], v[2] + v[5] == 10 v[1] + v[7], 10 v[1] + v[3] + v[10] == 10 v[1] + v[3]}

```

```

In[*]:= findVars[eq_] := Union@Cases[eq, _v, ∞]

```

```

In[*]:= makeNoGood[eq_] := Block[{vars},
  vars = findVars[eq];
  Or@@ (Not /@ Thread[x[vars, {##}]] &@@@ Cases[Tuples[Range[0, 9], Length@vars],
    x_ /; (eq /. Thread[vars -> x]) == False] // Apply[And]
]

```

```

In[*]:= bcf[spec_, vars_] := BooleanCountingFunction[spec, Length@vars] @@ vars;

In[*]:= unique = And @@ Table[bcf[{1}, Table[x[t, i], {i, 0, 9}]], {t, Values@imgToVar}] &&
  And @@ Table[bcf[{1}, Table[x[t, i], {t, Values@imgToVar}]], {i, 0, 9}];

In[*]:= bf = And @@ (makeNoGood /@ eqs) && unique;

In[*]:= vars = BooleanVariables[bf];

In[*]:= inst = SatisfiabilityInstances[bf, vars, All, Method -> "SAT"]

Out[*]=
{{False, True, False, False, False, False, False, False, False, False, False, False,
  False, False, True, False, False, False, False, False, False, False, False, False,
  True, False, False, False, False, False, False, True, False, False, False, False, False,
  False, False, False, False, False, False, False, False, True, False, False, False,
  False, False, False, True, False, False, False, False, False, False, False, True, False,
  False, False, False, False, False, False, False, False, False, False, False, False,
  False, False, True, False, False, False, False, False, False, False, False, False,
  False, True, True, False, False, False, False, False, False, False, False, False}}

In[*]:= Cases[Thread[vars -> inst[[1]]], HoldPattern[x[a_, b_] -> True] :> varToImg[a] -> b] //
  SortBy[Last] // TableForm

```

Out[*]//TableForm=

