# **Corticon Rule Modeling Challenge – Professor Smart's Safe**

## **Professor Smart's Safe**

Professor Smart has locked his lunch in his safe and forgotten the nine digit combination. However he does recall that the digits in the combination (abcdefghi) are all different and conform to these rules: d - f = ga \* b \* c < h \* ib + c + f < ha = b + ia = 4h > 6i < hCan you design a decision model that will help Professor Smart get his lunch?



## **The Approach**

The basic idea is to use a brute force approach by employing Corticon's scope section to generate (all) possible codes and then use the rules (actually just one) to check each code to see if it conforms to the conditions.

### Vocabulary



The FiniteDomain entity is used to define the digits 1,2,3,4,5,6,7,8,9

#### **Rules**

ProfSmart_Solve.ers 🗙				
Scope		Conditions	0	1
E. FiniteDomain [a]	aj	a<>i		Т
	ak	d.value - ef.value = g.value		Т
E FiniteDomain [c]	al	a.value*b.value*c.value < h.value * i.value		Т
	am	b.value+c.value+ef.value < h.value		Т
🗄 🔚 FiniteDomain [d]	an	a.value = b.value + i.value		Т
🗄 🔚 FiniteDomain [e]	ao	a.value = 4		Т
F. EiniteDomain [ef]	ар	h.value > 6		Т
	aq	i.value < h.value		Т
	ar			
🖭 🚍 FiniteDomain [h]	as			
🗄 🔚 FiniteDomain [i]	at			

The nine aliases to the Entity called FiniteDomain enable us to try evry combination of nine things.

The possible values the nine things can take will be set up in the test data – it this case it will be the nine digits 1-9. But we could equally well give Corticon any nine numbers an ask it to find a solution.

Conditions a thru ai are not shown but simply check to make sure that the digits are all different – similar to aj. Some constraint languages employ an "allDifferent" function which Corticon does not have. (if you disable these conditions then Corticon will find solutions that include repeated digits)

#### **Test Case**

