Setup finite fields

In[1]:=
```
<< FiniteFields` (* Load finite field package *)
SetFieldFormat[GF[2, 2], FormatType → FunctionOfCode[F]]
(* Set F as field with 4 elements *)
Output[x_] := x /. {F[1] → 1, F[2] → A, F[3] → B};
(* Function to make field elements {0,1,A,B} *)
```

Addition and multiplication tables:

In[4]:=
```
elements = {F[0], F[1], F[2], F[3]};
fullElements = {elements, elements, elements, elements};
plus = fullElements + Transpose[fullElements];
multiply = fullElements * Transpose[fullElements];
plusTab = ArrayFlatten[{{"+", {elements}}, {Transpose[{elements}], plus}}] // Output;
multiplyTab =
  ArrayFlatten[{{"*", {elements}}, {Transpose[{elements}], multiply}}] // Output;
Grid[plusTab, Alignment → Left, Spacings → {2, 1}, Frame → All,
 ItemStyle → "Text", Background → {{Gray, None}, {LightGray, None}}]
Grid[multiplyTab, Alignment → Left, Spacings → {2, 1}, Frame → All,
 ItemStyle → "Text", Background → {{Gray, None}, {LightGray, None}}]
```

Out[10]=

+	0	1	A	B
0	0	1	A	B
1	1	0	B	A
A	A	B	0	1
B	B	A	1	0

Out[11]=

*	0	1	A	B
0	0	0	0	0
1	0	1	A	B
A	0	A	B	1
B	0	B	1	A

Multivariate constrained expression:

Constraints : u(0,y) = A, u(B,y) = 1, u(x,0) = u(x,B)

In[12]:=
```
s1[x_] := {1, x};
Sij = {s1[0], s1[F[3]]};
α1 = Inverse[Sij];
ϕ1[x_] := s1[x].α1;
ρ1[x_, y_, g_Symbol] := {F[2] - g[0, y], 1 - g[F[3], y]};
u1[x_, y_, g_Symbol] := g[x, y] + ϕ1[x].ρ1[x, y, g];
ρ1[x, y, h] // Output
```

Out[18]=
```
{A - h[0, y], 1 - h[B, y]}
```

In[19]:=
```
s2[y_] := {y};
Sij = {s2[0] - s2[F[3]]};
α2 = Inverse[Sij];
ϕ2[y_] := s2[y].α2;
ρ2[x_, y_, g_Symbol] := {u1[x, F[3], g] - u1[x, 0, g]};
u[x_, y_, g_Symbol] := u1[x, y, g] + ϕ2[y].ρ2[x, y, g];
```

Check constraints for any g (x):

In[25]:=
```
FullSimplify[u[0, y, g] == F[2]]
FullSimplify[u[F[3], y, g] == 1]
FullSimplify[u[x, 0, g] == u[x, F[3], g]]
```

Out[25]=
```
True
```

Out[26]=
```
True
```

Out[27]=
```
True
```

Check constraints for a specific g(x):

In[28]:=
```
(*g[x_,y_]:=x*x;*)
g[x_, y_] := F[2] * x + x * y + y;
uVals = Table[u[F[x], F[y], g], {y, 3, 0, -1}, {x, 0, 3}];
uTab =
  ArrayFlatten[{{Transpose[{Reverse[elements]}], uVals}, {"y/x", {elements}}}] // Output;
Grid[uTab, Alignment → Left, Spacings → {2, 1}, Frame → All, ItemStyle → "Text",
 Background → {{Gray, None}, {None, None, None, None, Gray}}]
```

Out[31]=

B	A	B	0	1
A	A	B	0	1
1	A	B	0	1
0	A	B	0	1
y/x	0	1	A	B