Demo markov chain

last edited May 1, 2013 6:02:29 PM by admin

We implement the action of the operators:

```
Omega = CartesianProduct([0,1],[0,1]).map(tuple)
```

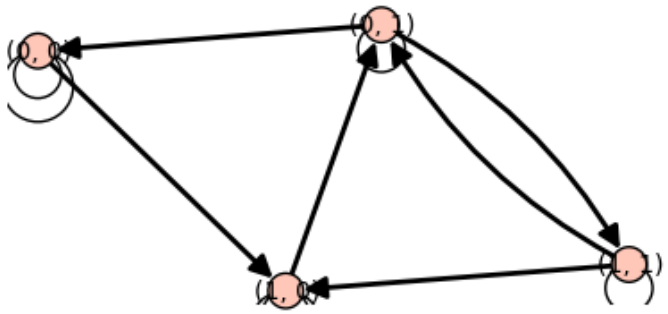
evaluate

```
def source(l, i=0):
      result = copy(l)
      for j in range(i,len(result)):
          if l[i] == 0: return l[:i]+ (1,) + l[i+1:]
      return l
```

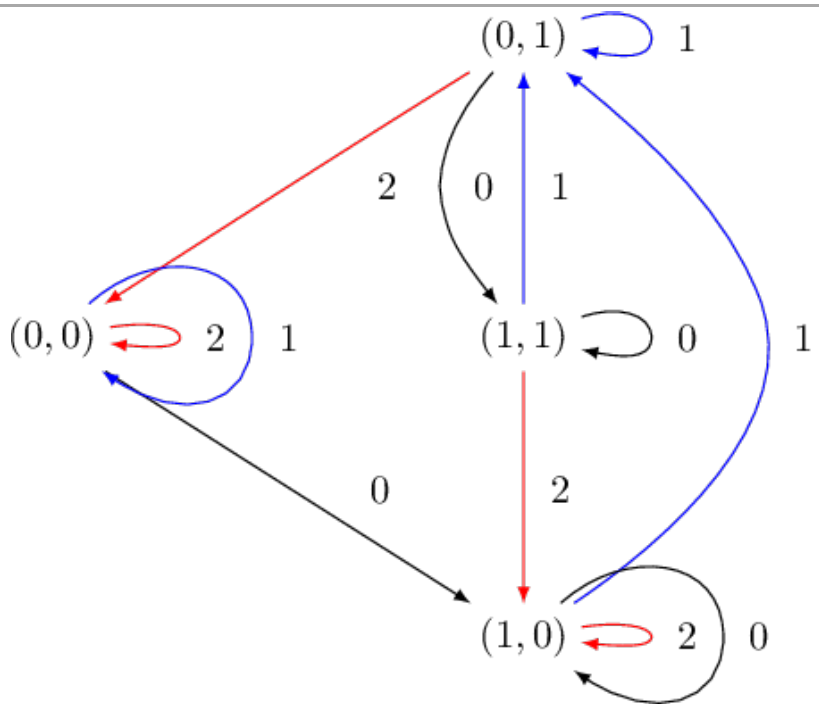
```
def topple(l, i):
      if l[i] == 0: return l
      return source(l[:i] + (0,) + l[i+1:], i+1)
```

From those, we build the graph of the Markov chain:

```
G = DiGraph(multiedges=True, loops=True)
G.add_vertices(Omega)
for c in Omega:
      G.add_edge(c, source(c), 0)
      G.add_edge(c, topple(c,0), 1)
      G.add_edge(c, topple(c,1), 2)
G.show()
```



```
G.set_latex_options(format="dot2tex", edge_labels=True,
color_by_label=CartanType.color)
view(G, tightpage=True)
```




```
G.is_strongly_connected()
```

```
True
```

We now consider Omega endowed with the action of the monoid generated by the operators:

```
V = G.transition_module(side="left"); V
```

```
{(0, 0), (0, 1), (1, 0), (1, 1)} endowed with an action of The
transition monoid of Looped multi-digraph on 4 vertices
```

```
M = V.semigroup(); M
```

```
The transition monoid of Looped multi-digraph on 4 vertices
```

```
M.cardinality()
```

```
11
```

```
M.is_l_trivial()
```

```
False
```

```
M.is_r_trivial()
```

```
True
```

We recover the eigenvalues using character theory:

```
M = G.transition_monoid(side="left", category=RTrivialMonoids().Finite())
V = G.transition_module(side="left", monoid=M); V
KV = V.algebra(QQ)
```

```
KV.character()
```

```
4*C[()] + 2*C[(0,)] + C[(0, 1, 2)] + 2*C[(1,)] + 2*C[(2,)]
```

```
KV.composition_factors()
```

```
S[(0,)] + S[(0, 1, 2)] + S[(1,)] + S[(2,)]
```

Internally, this uses the *character table* of the monoid, encoded as a change of basis in the character ring:

```
G0 = M.character_ring()
for s in G0.S().basis():
      print G0.C()(s)
```

```
C[()]
C[()] + C[(2,)]
C[()] + C[(0,)]
C[()] + C[(1,)]
C[()] + C[(0,)] + C[(0, 1, 2)] + C[(1,)] + C[(2,)]
```

```

```