Conto Flow

Michael.Schreiber

Thu 31 May 2018 17:34:55

Goals

1. Distinguish traditional transfer events from non traditional flows between contos.
2. Simulate blockchains which support particular protocols for related features.
3. Find concise specifications which simplify verifications of implementations.

Approach

Changes of conto balances are recorded with block time stamps according to rates of flows.
Types of contos are defined upon creation to model players and supply is fixed by total initial balances.
Flows with arbitrary rates change particular effects of distribution but conserve total balances.
Accumulated results of flows are recorded before changing rates.
Accumulated rates are distributed among destinations of flows.

Install Package

In[•]:=
```
Needs["ContoFlow`"]
```

In[•]:=
```
<< ContoFlow`
```

In[•]:=
```
Remove[Evaluate["ContoFlow`" <> "*"]]
```

Two Conto Flow Example

In[•]:=
```
{BlockTime[], Contos[], Flows[]}
```

Out[•]=
```
{0, ‹| |›, ‹| |›}
```

New Contos

In[•]:=
```
NewConto["a1", 0, 1000]; NewConto["a2", 1, 10 000]; Dataset[Contos[]]
```

Out[•]=

	contyp	ctt	bala	bat	ifis	ift	ofis	oft
a1	0	0	1000	0		0		0
a2	1	0	10 000	0		0		0

New Flow

In[•]:=
```
NewFlow["a1", "a2", 100]; Dataset[Flows[]]
```

Out[•]=

a1.a2	accu	0
	act	0
	rate	100
	rat	0

Harvest

In[•]:=
```
BlockTimePlus[]
```

Out[•]= 1

In[•]:=
```
Harvest["a1.a2"]; Column[{Dataset[Contos[]], Dataset[Flows[]]}]
```

	contyp	ctt	bala	bat	ifis	ift	ofis	oft
a1	0	0	900	1	{}	0	{a1.a2}	0
a2	1	0	10100	1	{a1.a2}	0	{}	0

Out[•]=

a1.a2	accu	0
	act	1
	rate	100
	rat	0

Pay

In[•]:=
```
Pay["a2", "a1", 10]
```

In[•]:=
```
Column[{Dataset[Contos[]], Dataset[Flows[]]}]
```

	contyp	ctt	bala	bat	ifis	ift	ofis	oft
a1	0	0	910	1	{}	0	{a1.a2}	0
a2	1	0	10090	1	{a1.a2}	0	{}	0

Out[•]=

a1.a2	accu	0
	act	1
	rate	100
	rat	0

Fast Forward

In[•]:=
```
BlockTimePlus[11]
```

Out[•]=
```
12
```

In[•]:=
```
Harvest["a1.a2"]
```

In[•]:=
```
Column[{Dataset[Contos[]], Dataset[Flows[]]}]
```

	contyp	ctt	bala	bat	ifis	ift	ofis	oft
a1	0	0	0	12	{}	0	{a1.a2}	0
a2	1	0	11000	12	{a1.a2}	0	{}	0

Out[•]=

a1.a2	accu	190
	act	12
	rate	100
	rat	0

In[•]:=
```
Pay["a2", "a1", 200]
```

In[•]:=
```
Column[{Dataset[Contos[]], Dataset[Flows[]]}]
```

	contyp	ctt	bala	bat	ifis	ift	ofis	oft
a1	0	0	200	12	{}	0	{a1.a2}	0
a2	1	0	10800	12	{a1.a2}	0	{}	0

Out[•]=

a1.a2	accu	190
	act	12
	rate	100
	rat	0

In[•]:=
```
Harvest["a1.a2"]
```

In[•]:=
```
Column[{Dataset[Contos[]], Dataset[Flows[]]}]
```

	contyp	ctt	bala	bat	ifis	ift	ofis	oft
a1	0	0	10	12	{}	0	{a1.a2}	0
a2	1	0	10990	12	{a1.a2}	0	{}	0

Out[•]=

a1.a2	accu	0
	act	12
	rate	100
	rat	0

100 contos

In[•]:=
```
ContoFlowReset[]
```

In[•]:=
```
Timing[Table[NewConto[ToString[n], Mod[n, 2], 10], {n, 100}];]
```

Out[•]=
```
{0.000932, Null}
```

In[•]:=
```
Dataset[Contos[]]
```

Out[•]=

	contyp	ctt	bala	bat	ifis	ift	ofis	oft
1	1	0	10	0		0		0
2	0	0	10	0		0		0
3	1	0	10	0		0		0
4	0	0	10	0		0		0
5	1	0	10	0		0		0
6	0	0	10	0		0		0
7	1	0	10	0		0		0
8	0	0	10	0		0		0
9	1	0	10	0		0		0
10	0	0	10	0		0		0
11	1	0	10	0		0		0
12	0	0	10	0		0		0
13	1	0	10	0		0		0
14	0	0	10	0		0		0
15	1	0	10	0		0		0
16	0	0	10	0		0		0
17	1	0	10	0		0		0
18	0	0	10	0		0		0
19	1	0	10	0		0		0
20	0	0	10	0		0		0

showing 1–20 of **100**

In[•]:=
```
Timing[Table[Table[
    With[{rc = RandomInteger[{1, n}]}, If[rc == n, Nothing, NewFlow[ToString[n],
        ToString[rc], RandomInteger[{1, 10}]]]];, {m, 20}], {n, 1, 100}];]
```

Out[•]=
```
{0.050722, Null}
```

In[•]:= **Dataset[Flows[]]**

Out[•]=

	accu	act	rate	rat
2.1	0	0	9	0
3.1	0	0	4	0
3.2	0	0	5	0
4.1	0	0	3	0
4.2	0	0	1	0
4.3	0	0	9	0
5.3	0	0	8	0
5.2	0	0	6	0
5.1	0	0	8	0
5.4	0	0	1	0
6.3	0	0	8	0
6.5	0	0	2	0
6.2	0	0	9	0
6.4	0	0	2	0
6.1	0	0	4	0
7.4	0	0	7	0
7.6	0	0	6	0
7.1	0	0	4	0
7.5	0	0	4	0
7.3	0	0	2	0

showing 1–20 of **1447**

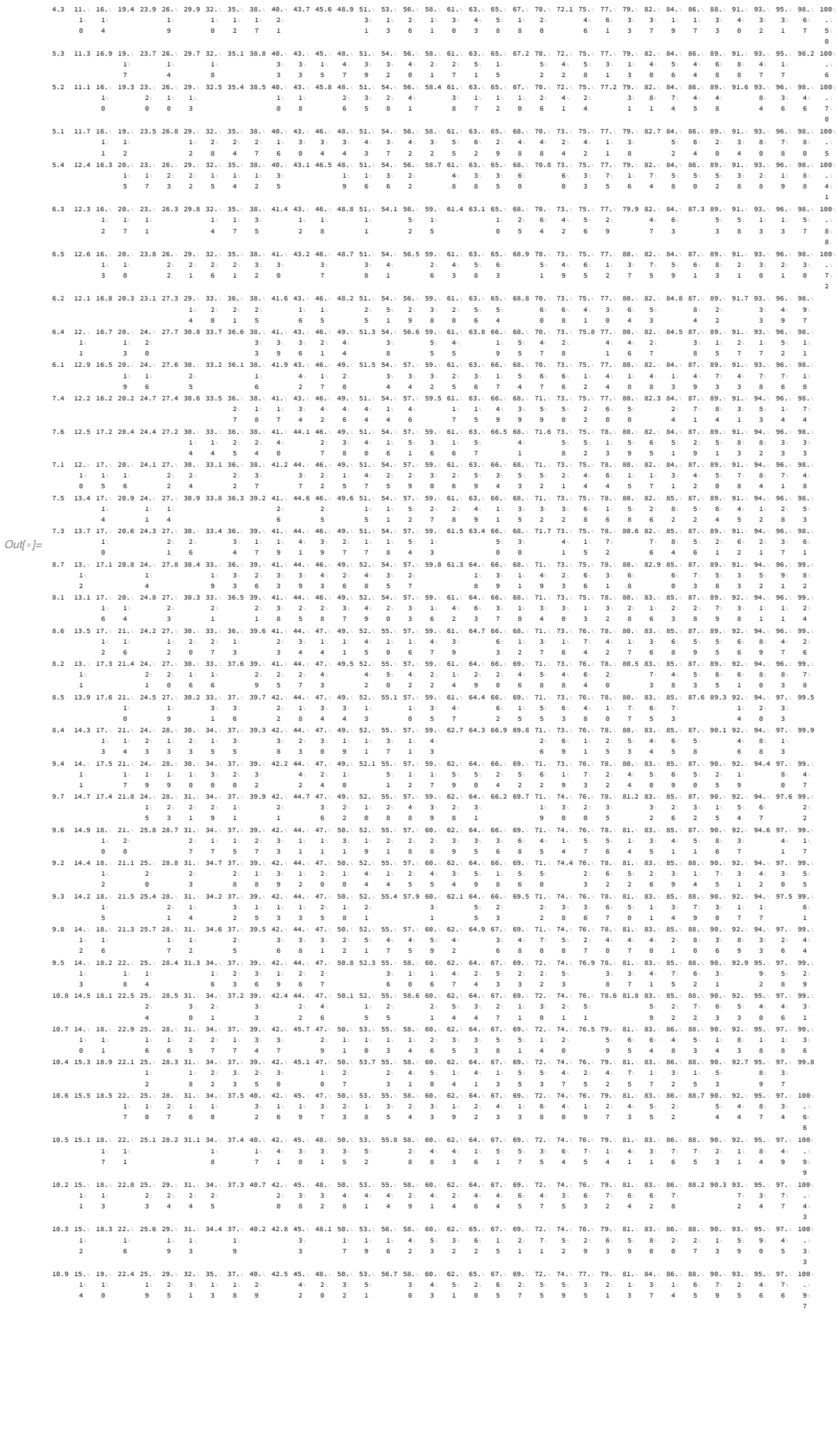
In[•]:= **Table[Contos[ToString[c], "bala"], {c, 100}]**

Out[•]= {10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,
10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,
10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,
10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,
10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10}

In[•]:= **Multicolumn[Keys@Flows[], 36, BaseStyle → {FontSize → 5}]**

```
2.1  10.1 15.8 19.2 22.\ 26.4 29.\ 32.\ 35.7 37.8 40.\ 43.\ 45.\ 48.\ 50.\ 53.\ 56.\ 58.\ 60.\ 62.\ 65.1 67.7 70.7 72.\ 74.\ 77.\ 79.\ 81.\ 84.\ 86.\ 88.\ 90.\ 93.\ 95.\ 97.\ 100\
                    1\       1\   1\          2\   2\   3\   1\   4\   3\   1\   2\   1\   2\               2\   4\   4\   3\   2\   7\   6\   7\   4\   5\   5\   9\   .\
                    5        7    3           2    2    2    8    4    6    0    6    8    0               0    3    9    4    3    5    1    6    6    3    0    2    2\
                                                                                                                                                                  0
3.1  11.6 15.4 19.\ 22.\ 26.\ 29.\ 32.\ 35.\ 38.\ 40.\ 43.\ 45.\ 48.\ 50.\ 53.\ 56.\ 58.\ 60.\ 63.\ 65.\ 67.\ 70.2 72.\ 74.\ 77.\ 79.\ 81.\ 84.\ 86.\ 88.\ 90.9 93.\ 95.\ 98.\ 100\
                1\   2\   2\   2\   2\   2\   2\   2\   1\   4\   4\   3\   2\   2\   4\   3\   4\   5\   3\        5\   3\   2\   4\   7\   3\   6\   5\        5\   2\   8\   .\
                6    1    1    7    3    5    1    7    1    3    0    3    2    5    7    9    1    3    4         2    7    5    3    8    4    0    0         4    3    3    4\
                                                                                                                                                                  7
3.2  11.4 15.2 19.\ 23.\ 26.\ 29.\ 32.\ 35.\ 38.2 40.\ 43.\ 45.\ 48.6 50.2 53.9 56.\ 58.\ 60.\ 63.\ 65.\ 67.\ 70.\ 72.\ 74.\ 77.\ 79.\ 81.\ 84.\ 86.\ 88.\ 90.\ 93.\ 95.\ 98.\ 100\
                1\   2\   2\   1\   2\   2\        1\   2\   3\                  3\   3\   1\   1\   2\   1\   3\   4\   2\   1\   7\   4\   3\   3\   7\   2\   2\   8\   9\   .\
                5    2    4    6    0    8         2    0    4                   9    5    2    9    3    9    7    0    4    6    0    4    8    6    0    8    7    0    2    9\
                                                                                                                                                                  2
4.1  11.8 16.1 19.5 23.\ 26.6 29.1 32.\ 35.\ 38.\ 40.1 43.\ 45.\ 48.\ 51.\ 53.\ 56.\ 58.\ 60.\ 63.\ 65.6 67.\ 70.\ 72.\ 74.\ 77.\ 79.\ 82.\ 84.\ 86.\ 88.\ 91.\ 93.\ 95.\ 98.\ 100\
                    1\             1\   1\   1\        1\   2\   2\   2\   2\   4\   3\   3\   4\        5\   4\   1\   2\   5\   4\   6\   2\   8\   8\   2\   5\   4\   4\   .\
                    3              1    6    9         9    6    9    1    4    4    3    7    6         6    2    0    3    5    8    1    3    2    1    3    0    2    2    2\
                                                                                                                                                                  3
4.2  11.9 16.4 19.\ 23.\ 26.2 29.\ 32.\ 35.5 38.\ 40.\ 43.\ 45.\ 48.\ 51.\ 53.\ 56.1 58.\ 60.\ 63.\ 65.8 67.\ 70.\ 72.\ 75.\ 77.\ 79.5 82.\ 84.\ 86.\ 88.\ 91.\ 93.\ 95.9 98.\ 100\
                1\   1\        2\   2\        2\   3\   3\   1\   1\   2\   1\        5\   1\   3\        4\   5\   2\   4\   3\        6\   3\   3\   4\   2\   3\        7\   .\
                1    0         8    1         6    0    2    1    6    4    6         0    0    4         7    4    4    0    8         8    3    0    9    5    4         2    5\
                                                                                                                                                                  7
```

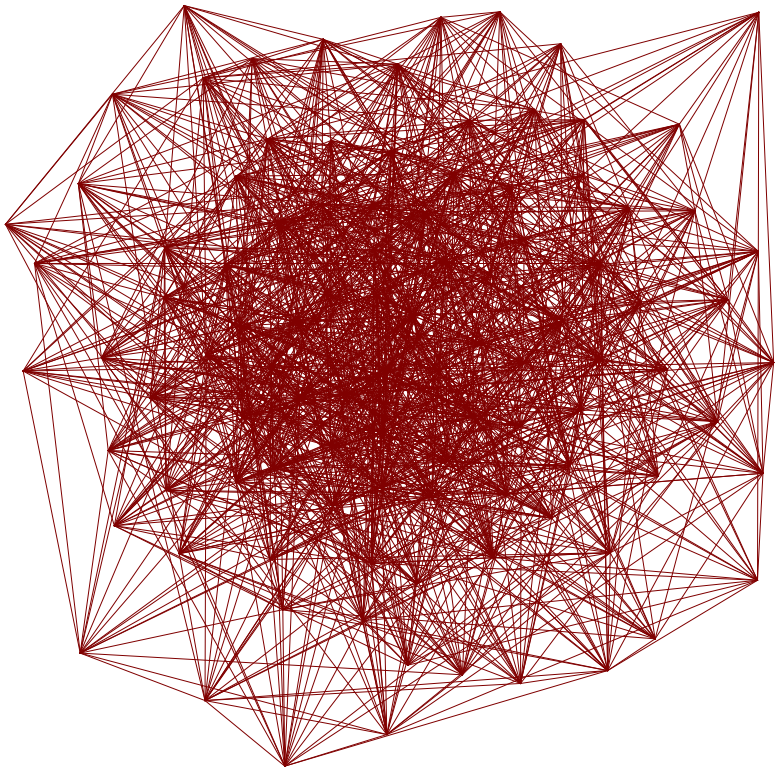
Out[•]=



Graph of Flows

In[•]:=
```
GraphPlot[Rule @@@ (StringSplit[#, "."] & /@ Keys[Flows[]]),
 ImageSize → 500, VertexRenderingFunction → None]
```

Out[•]=



In[•]:=
```
BlockTimePlus[]
```

Out[•]=
```
1
```

In[•]:=
```
Timing[Harvest[#] & /@ Keys[Flows[]];]
```

Out[•]=
```
{0.07798, Null}
```

In[•]:=
```
Table[Contos[ToString[c], "bala"], {c, 100}]
```

Out[•]=
$$\left\{\frac{2\,393\,575\,482\,002\,880\,897\,049\,159\,162}{41\,113\,491\,775\,332\,745\,943\,910\,645},\ \frac{21\,648\,000\,239\,746\,955\,542\,487\,741\,768\,729}{597\,714\,885\,111\,885\,622\,539\,410\,092\,140},\right.$$
$$\frac{5\,650\,049\,173\,814\,842\,250\,172\,364\,699}{152\,155\,888\,888\,856\,369\,382\,585\,720},\ \frac{590\,493\,574\,595\,718\,530\,619\,576\,641}{21\,098\,637\,131\,784\,849\,733\,216\,120},$$
$$\frac{1\,817\,475\,328\,446\,693\,074\,493\,166\,095\,191}{54\,313\,399\,351\,586\,565\,388\,649\,571\,120},\ \frac{155\,991\,717\,884\,572\,896\,019\,838\,321\,483}{6\,117\,775\,893\,187\,429\,160\,678\,869\,881},$$
$$\frac{2\,584\,334\,702\,904\,509\,777\,781\,228\,959}{87\,459\,147\,041\,328\,772\,289\,636\,238},\ \frac{102\,544\,755\,988\,695\,221\,911\,309}{3\,940\,307\,790\,596\,605\,091\,664},$$
$$\frac{704\,796\,719\,358\,231\,179\,815\,639\,519}{31\,772\,589\,785\,858\,703\,624\,737\,895},\ \frac{15\,425\,785\,235\,820\,337\,234\,885\,673}{790\,257\,482\,379\,499\,006\,228\,356},$$

$$\left\{\frac{69\,656\,979\,729\,857\,060\,244\,536\,835}{3\,132\,947\,746\,860\,733\,531\,746\,926}, \frac{4\,608\,252\,849\,598\,173\,368}{253\,898\,087\,162\,658\,939}, \frac{868\,300\,011\,904\,141\,275\,167}{38\,723\,655\,151\,158\,125\,148}, \frac{479\,520\,365\,589\,919\,405\,672\,075}{25\,625\,344\,579\,309\,650\,294\,504}, \frac{122\,841\,393\,944\,812\,244\,276\,029\,139}{6\,832\,792\,013\,598\,231\,715\,844\,964}, \frac{4\,907\,377\,836\,823\,759\,511\,862\,620\,347}{214\,782\,712\,380\,340\,184\,260\,652\,472}, \frac{16\,571\,475\,083\,879\,112\,602\,094\,081\,380\,861}{782\,968\,414\,712\,820\,566\,130\,887\,850\,864}, \frac{774\,542\,365\,466\,834\,198\,581}{58\,492\,246\,658\,279\,516\,082}, \frac{728\,065\,064\,970\,030\,647\,003}{39\,039\,740\,292\,322\,001\,232}, \frac{9\,650\,291\,137\,745\,006\,105\,473}{563\,126\,384\,138\,415\,401\,304}, \frac{4\,452\,604\,998\,076\,872\,650\,583\,457}{232\,296\,515\,682\,522\,576\,729\,444}, \frac{28\,928\,103\,753\,080\,481\,140\,183}{1\,664\,872\,489\,805\,403\,482\,604}, \frac{9\,068\,308\,022\,952\,572\,690\,502\,781}{599\,838\,638\,209\,332\,491\,429\,460}, \frac{6\,689\,059\,251\,678\,146\,891}{536\,539\,354\,004\,109\,456}, \frac{97\,225\,857\,002\,811\,286\,549\,666\,764\,013}{5\,870\,584\,942\,452\,284\,332\,922\,179\,536}, \frac{41\,740\,716\,142\,113\,376\,703\,059}{3\,336\,478\,695\,006\,056\,511\,912}, \frac{665\,362\,111\,845\,587\,836\,278\,306\,479}{41\,132\,776\,216\,556\,591\,992\,689\,264}, \frac{1\,358\,700\,704\,776\,057\,413\,791}{128\,009\,171\,242\,574\,748\,126}, \frac{329\,278\,256\,741\,142\,009\,779}{43\,607\,361\,030\,280\,929\,882}, \frac{17\,137\,854\,014\,694\,978\,275}{1\,468\,622\,585\,362\,729\,104}, \frac{4\,445\,166\,763\,422\,181\,549}{417\,827\,697\,599\,576\,880}, \frac{453\,523\,121\,461\,462\,797\,739}{48\,499\,328\,083\,175\,417\,448}, \frac{2\,278\,966\,204\,467\,470\,400\,084\,155}{140\,265\,835\,183\,610\,780\,635\,992}, \frac{256\,520\,999\,055\,050\,021\,269}{13\,706\,002\,693\,740\,122\,766}, \frac{2\,612\,808\,300\,528\,022\,433\,351}{248\,750\,427\,905\,282\,677\,374}, \frac{4\,353\,735\,799\,865\,435}{408\,677\,134\,231\,548}, \frac{1\,304\,033\,699\,003\,513\,898\,339}{93\,906\,750\,594\,229\,869\,296}, \frac{678\,087\,845\,372\,610\,675}{67\,913\,925\,071\,141\,968}, \frac{380\,169\,256\,919\,593\,901}{47\,280\,198\,147\,087\,906}, \frac{1\,635\,680\,661\,361\,883}{265\,212\,348\,884\,490}, \frac{254\,467\,777\,506\,278\,627\,251\,589}{22\,281\,460\,461\,216\,381\,300\,876}, \frac{58\,250\,289\,533\,287\,851\,343}{5\,361\,880\,290\,778\,806\,795}, \frac{359\,483\,156\,980\,229}{35\,577\,130\,217\,706}, \frac{20\,740\,219\,723\,119\,601}{1\,875\,384\,936\,137\,097}, \frac{483\,817\,065}{99\,085\,112}, \frac{1\,098\,302\,303\,411}{197\,852\,883\,844}, \frac{103\,554\,221\,616\,800\,336\,707}{9\,649\,342\,373\,830\,662\,888}, \frac{73\,669\,446\,978\,982\,512\,563}{8\,476\,224\,972\,542\,858\,916}, \frac{90\,823\,115\,200\,871}{13\,214\,971\,495\,089}, \frac{10\,053\,347\,612\,021\,306\,423}{1\,223\,238\,368\,994\,693\,480}, \frac{33\,315\,576\,958\,156\,228\,279}{3\,233\,129\,331\,397\,453\,488}, \frac{15\,830\,075\,250\,851}{2\,041\,814\,546\,408}, \frac{785\,970\,236\,088\,431}{76\,004\,537\,421\,072}, \frac{6\,558\,830\,593}{1\,598\,034\,357}, \frac{388\,090\,018\,133}{66\,457\,666\,296}, \frac{79\,627\,101}{19\,558\,448}, \frac{730\,889\,944}{253\,712\,909}, \frac{6\,180\,664\,488\,065}{821\,533\,409\,424}, \frac{83\,406\,045\,033\,817}{12\,077\,930\,184\,684}, \frac{493\,249\,595}{160\,831\,044}, \frac{553\,875\,002\,779}{129\,883\,653\,198}, \frac{7\,997\,337\,199}{1\,467\,827\,634}, \frac{902\,000\,915}{258\,030\,864}, \frac{8\,538\,336\,679}{1\,502\,698\,974}, \frac{967\,493\,782\,565}{279\,458\,391\,366}, \frac{33\,633}{11\,186}, \frac{16\,277\,035}{5\,163\,914}, \frac{421\,397\,376\,797}{68\,395\,619\,754}, \frac{755}{594}, \frac{2\,612\,145}{950\,011}, \frac{84\,393}{47\,242}, \frac{8\,269\,425}{3\,129\,448}, \frac{20\,084\,923}{6\,699\,693}, \frac{6\,725\,265}{3\,285\,584}, \frac{299\,242\,679\,695}{78\,535\,314\,352}, \frac{81\,858\,375}{22\,935\,359}, \frac{120}{97}, \frac{897\,681\,695}{288\,006\,192}, \frac{149\,097}{91\,471}, \frac{10\,942\,984\,973}{2\,228\,233\,560}, \frac{2735}{2993}, \frac{6\,607\,070}{1\,987\,821}, \frac{1\,656\,655}{895\,781}, \frac{1\,978\,487}{789\,360}, 0, \frac{40}{97}, 1, \frac{68\,990}{31\,349}, \frac{553}{391}, \frac{7}{10}, \frac{100}{79}, \frac{629\,827}{264\,422}, 0, 0, 0, 0, \frac{2335}{1363}, 0, \frac{45}{47}, 0\right\}$$

In[•]:=
```
Total@Table[Contos[ToString[c], "bala"], {c, 100}]
```

Out[•]=
```
1000
```

```
In[•]:= Table[With[{is = Contos[ToString[k], "ifis"], os = Contos[ToString[k], "ofis"],
    h = Hint[ToString[k]]}, If[EvenQ[k], h, If[Length[is] > 0, RandomChoice[is],
     If[Length[os] > 0, RandomChoice[os], 0]]]], {k, Range[100]}]
```

```
Out[•]= {49.1, 72.2, 8.3, 23.4, 64.5, 48.6, 61.7, 17.8, 16.9, 90.10, 52.11, 54.12,
 62.13, 73.14, 28.15, 70.16, 25.17, 35.18, 84.19, 92.20, 82.21, 54.22, 63.23,
 37.24, 26.25, 50.26, 42.27, 50.28, 92.29, 58.30, 94.31, 89.32, 75.33, 53.34,
 61.35, 79.36, 61.37, 65.38, 53.39, 56.40, 50.41, 95.42, 45.43, 91.44, 67.45,
 65.46, 56.47, 73.48, 74.49, 55.50, 98.51, 100.52, 61.53, 62.54, 98.55,
 59.56, 67.57, 95.58, 69.59, 93.60, 93.61, 77.62, 85.63, 81.64, 87.65, 91.66,
 98.67, 73.68, 82.69, 84.70, 100.71, 100.72, 99.73, 92.74, 76.75, 84.76,
 86.77, 86.78, 96.79, 82.80, 82.81, 85.82, 83.5, 88.84, 97.85, 95.86, 99.87,
 93.88, 92.89, 100.90, 91.60, 100.92, 93.22, 97.95, 97.96, 100.97, 99.95}
```

```
Table[With[{is = Contos[ToString[k], "ifis"], os = Contos[ToString[k], "ofis"],
    h = Hint[ToString[k]]}, If[EvenQ[k], h, If[Length[is] > 0, RandomChoice[is],
     If[Length[os] > 0, RandomChoice[os], 0]]]], {k, Range[100]}]
```

Simulate all contos for 100 steps

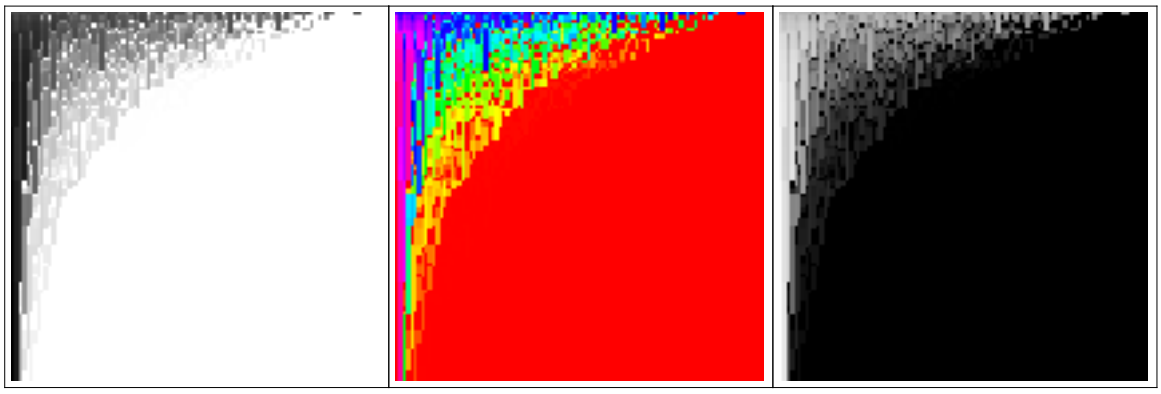
```
In[•]:= Timing[cf100100 = Table[(BlockTimePlus[];
      Table[Harvest[With[{is = Contos[ToString[k], "ifis"],
          os = Contos[ToString[k], "ofis"], h = Hint[ToString[k]]},
        If[EvenQ[k], h, If[Length[is] > 0, RandomChoice[is], If[Length[os] > 0,
          RandomChoice[os], 0]]]]], {k, RandomSample[Range[100]]}];
      Contos[ToString[#], "bala"] & /@ Range[100]), {s, 100}];]
```

```
Out[•]= {26.802, Null}
```

Plots of 100 account balances for 100 steps.

```
In[•]:= Row[{ArrayPlot[N[cf100100]^.1, ImageSize → 200],
   ArrayPlot[N[cf100100]^.1, ImageSize → 200, ColorFunction → Hue],
   ArrayPlot[N[cf100100]^.1, ImageSize → 200, ColorFunction → GrayLevel]}]
```

```
Out[•]=
```



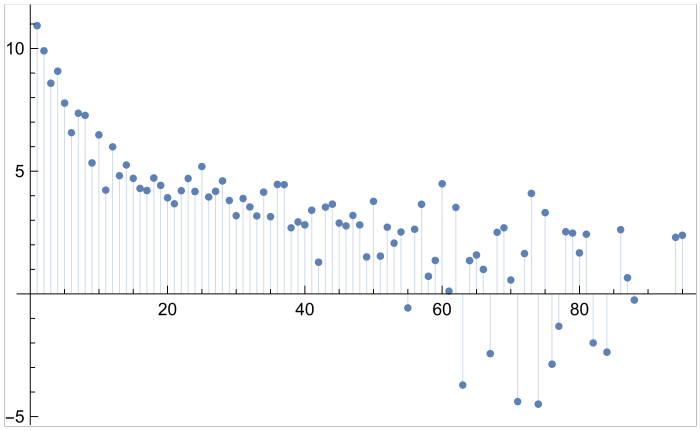
```
In[•]:= Total[Contos[ToString[#], "bala"] & /@ Range[100]]
```

```
Out[•]= 1000
```

```
cf100100
```

In[•]:=
```
ListPlot[Log[Total@cf100100], Filling → Axis]
```

Out[•]=



In[•]:=
```
Total /@ Transpose@Partition[Total@cf100100, 2]
```

Out[•]=
```
{6 456 063 536 842 807 839 032 428 881 466 022 256 334 824 064 124 449 279 948 748 885 827 920 ⁖
    586 867 985 581 388 199 559 650 484 636 964 538 430 059 202 307 690 449 237 753 603 000 786 ⁖
    158 566 843 /
   96 715 543 013 531 329 472 827 545 833 001 027 747 849 210 409 901 728 226 933 060 056 951 ⁖
    981 461 095 031 184 107 823 751 076 451 666 363 896 832 019 635 295 276 236 800 000 000 ⁖
    000 000 000,
  3 215 490 764 510 325 108 250 325 701 834 080 518 450 096 976 865 723 542 744 557 119 867 277 ⁖
    559 241 517 537 022 582 815 457 160 529 671 851 253 142 761 221 837 174 442 246 396 999 213 ⁖
    841 433 157 /
   96 715 543 013 531 329 472 827 545 833 001 027 747 849 210 409 901 728 226 933 060 056 951 ⁖
    981 461 095 031 184 107 823 751 076 451 666 363 896 832 019 635 295 276 236 800 000 000 ⁖
    000 000 000}
```

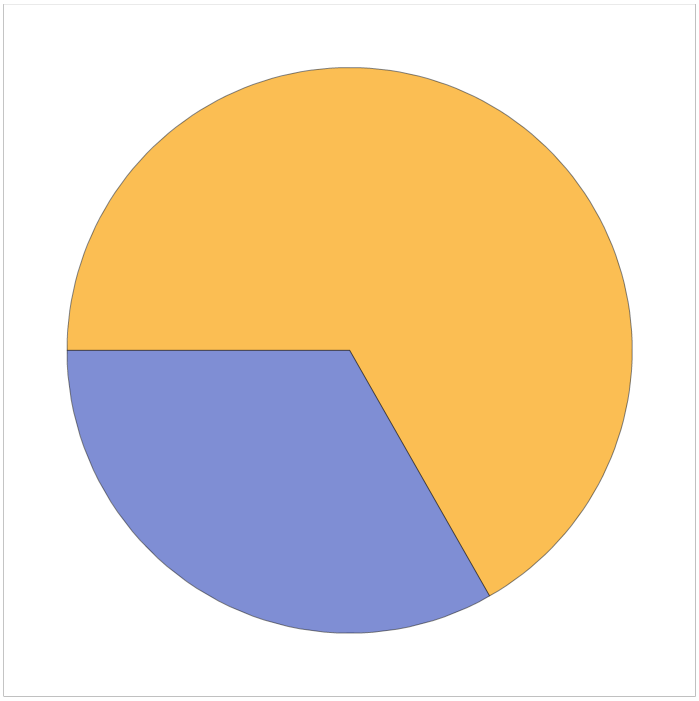
In[•]:=
```
N[Total /@ Transpose@Partition[Total@cf100100, 2]]
```

Out[•]=
```
{66 753.1, 33 246.9}
```

Even contos follw hints and get most of the pie.

In[•]:=
```
PieChart[Total /@ Transpose@Partition[Total@cf100100, 2]]
```

Out[•]=



In[•]:=
```
Total[Total /@ Transpose@Partition[Total@cf100100, 2]] / 100
```

Out[•]=
```
1000
```

In[•]:=
```
Row[{Dataset[Contos[]], Dataset[Flows[]]}]
```

Out[•]=

	contyp	ctt	bala	bat	ifis	ift
1	1	0	743.661	100	{ …37 }	0
2	0	0	256.319	100	{ …39 }	0
3	1	0	0.0199283	100	{ …41 }	0
4	0	0	0.00000000000000000000000000000251052	100	{ …40 }	0
5	1	0	0	100	{ …41 }	0
6	0	0	0.000000000105439	98	{ …37 }	0
7	1	0	0.00000000000000000000000000000195263	98	{ …47 }	0
8	0	0	0.00000000000000000000000000000351473	98	{ …33 }	0
9	1	0	0.0000000000000000000000000000015621	98	{ …30 }	0
10	0	0	0	98	{ …37 }	0
11	1	0	0	92	{ …35 }	0
12	0	0	0	91	{ …37 }	0
13	1	0	0	91	{ …34 }	0
14	0	0	0	89	{ …28 }	0
15	1	0	0	89	{ …31 }	0
16	0	0	0	81	{ …28 }	0
17	1	0	0	83	{ …23 }	0
18	0	0	0	81	{ …24 }	0
19	1	0	0	80	{ …33 }	0
20	0	0	0	80	{ …25 }	0

showing 1–20 of **100**

```
contoflowexamples
```

In[•]:=
```
Export["contoflowexport.json", <|
  "contoflows" -> <|"contos" -> Contos[], "flows" → Flows[]|>|>, "JSON"]
```

Out[•]=
```
contoflowexport.json
```

Export

In[•]:=
```
Export["contoflowexport.json", <|
  "contoflows" -> <|"contos" -> Contos[], "flows" → Flows[]|>|>, "JSON"]
```

Out[•]=
```
contoflowexport.json
```

10000 contos

In[•]:=
```
ContoFlowReset[]
```

In[•]:=
```
Timing[Table[NewConto[ToString[n], Mod[n, 2], 10], {n, 10 000}];]
```

Out[•]=
```
{0.110232, Null}
```

In[•]:=
```
Timing[Table[Table[
    With[{rc = RandomInteger[{1, n}]}, If[rc == n, Nothing, NewFlow[ToString[n],
      ToString[rc], RandomInteger[{1, 10}]]]];, {m, 20}], {n, 1, 10 000}];]
```

Out[•]=
```
{8.05389, Null}
```

Simulate 10000 contos for 100 steps assuming 100 harvests per step

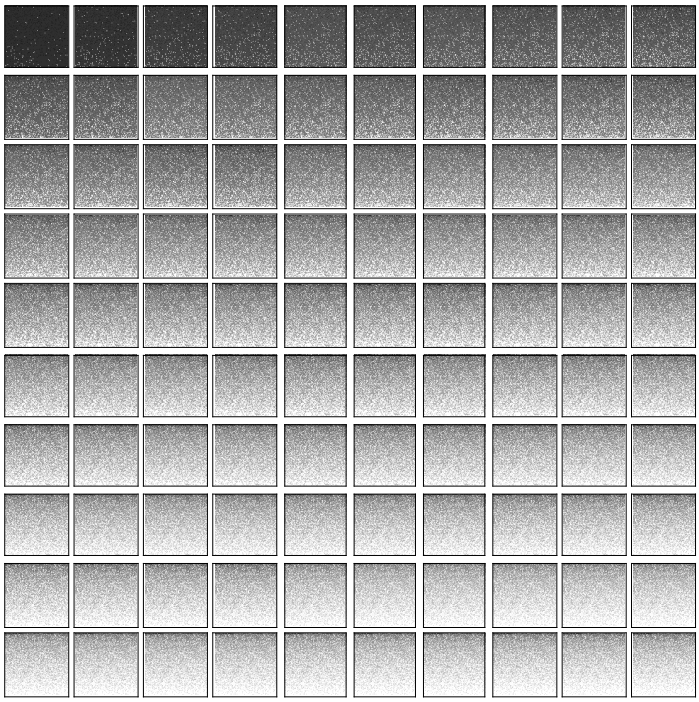
In[•]:=
```
Timing[cf1000100 = Table[(BlockTimePlus[];
    Table[Harvest[With[{is = Contos[ToString[k], "ifis"],
      os = Contos[ToString[k], "ofis"], h = Hint[ToString[k]]},
     If[EvenQ[k], h, If[Length[is] > 0, RandomChoice[is], If[Length[os] > 0,
       RandomChoice[os], 0]]]]], {k, RandomSample[Range[10 000], 100]}];
    Contos[ToString[#], "bala"] & /@ Range[10 000]), {s, 100}];]
```

Out[•]=
```
{46.6351, Null}
```

Plots of 10000 account balances in 100*100 array for 10 steps.

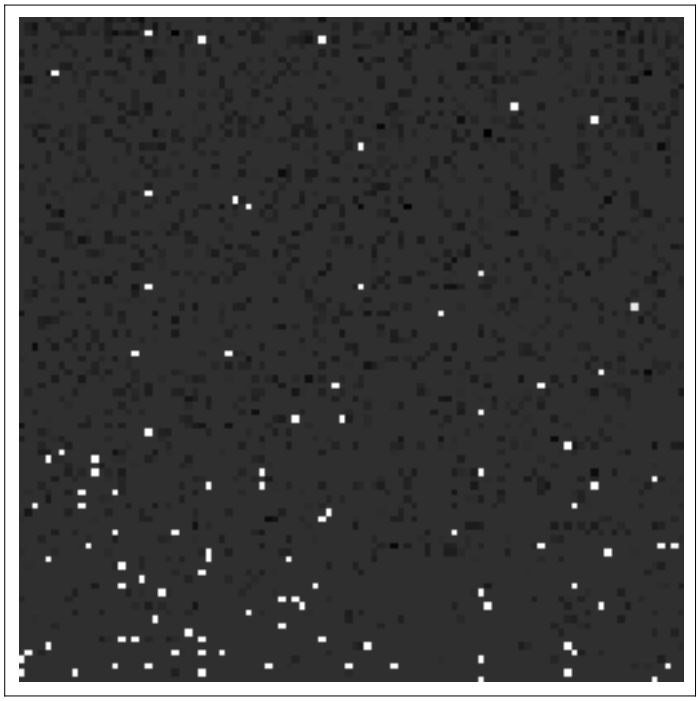
In[•]:=
```
GraphicsGrid[Partition[ArrayPlot[Partition[#, 100]] & /@ cf1000100, 10]]
```

Out[•]=



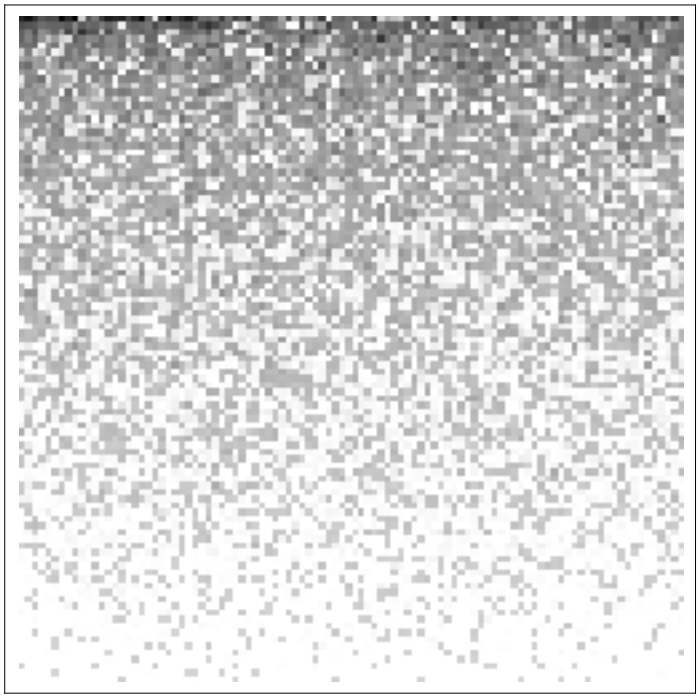
In[•]:=
```
ArrayPlot[Partition[cf1000100[[1]], 100]]
```

Out[•]=



In[•]:=
```
ArrayPlot[Partition[cf1000100[[100]], 100]]
```

Out[•]=



In[•]:=
```
Total[Contos[ToString[#], "bala"] & /@ Range[10 000]]
```

Out[•]=
```
100 000
```

Larger numbers of participants requesting harvests sporadically may reduce the effect of following hints as generated with heuristics implemented in this version.

In[•]:=
```
PieChart[Total /@ Transpose@Partition[Total@cf1000100, 2]]
```

Out[•]=