Apriori Algorithm

Theme

Data

In[13]:=
```
(* Data definition *)
x1 = {"Beer", "Nappies", "Tomatoes"};
x2 = {"Beer", "Nappies", "Crisps"};
x3 = {"Crisps"};
x4 = {"Beer", "Nappies", "Crisps"};

M = 4;
X = Table[xi, {i, 1, M}];(* List with all transactions *)
smin = 0.4;
kmin = 0.8;(* Reduce the rule output *)
```

Algorithm

Implementation

In[21]:=
```
(* Functions *)
support[Y_] := Count[X, i_ /; ContainsAll[i, Y]] / M;
(* Count each transaction which covers Y completely *)
confidence[Y_, Z_] := support[Y ⋃ Z] / support[Y];

step2CalcSupport[] := (
   (* Each element from Hn with it's support *)
   Sn = ({#, support[#]}) & /@ Hn // N;
  );
step3MinSupport[] := (
   (* Keep only elements from Hn which fulfill the support condition *)
   In = Cases[Hn, Y_ /; support[Y] ≥ smin];
   Print[ToString[Subscript["I", n], StandardForm] <> " = " <> ToString[In, StandardForm]];
  );
step4Merge[] := (
   𝓘 = 𝓘 ⋃ In;
   Print["𝓘 = " <> ToString[𝓘, StandardForm]];
  );
step5SizeIncrease[] := (
   (* Find next elements *)
   Hn+1 = Union[Sort[(* Keep a sorted list without duplicates *)
      Cases[
       Flatten[Outer[(Flatten@{#1 ⋃ #2}) &, I1, In, 1], 1],
       (* Cartesian cross product between In and I1 *)
       Y_ /; Length[Y] ≥ n + 1(* Keep only elements with proper length *)
      ]
     ]];
   Print[ToString[Subscript["H", n + 1], StandardForm] <> " = " <> ToString[Hn+1, StandardForm]];
  );
step6Increment[] := (
   ++n;
   Print["n = " <> ToString[n, StandardForm]];
  );
```

Step I

In[28]:=
```
n = 1;
𝓘 = {};
H1 = ({#}) & /@ Union[Flatten[X, 1]]
```

Out[30]=
```
{{Beer}, {Crisps}, {Nappies}, {Tomatoes}}
```

In[31]:=
```
While[True,
 step2CalcSupport[];
 step3MinSupport[];
 If[Length[In] == 0,
  Print["Algorithm terminated"];
  Break[]
 ];
 step4Merge[];
 step5SizeIncrease[];
 step6Increment[];
]
```

$\mathbb{I}_1$ = {{Beer}, {Crisps}, {Nappies}}

$\mathcal{I}$ = {{Beer}, {Crisps}, {Nappies}}

H_2 = {{Beer, Crisps}, {Beer, Nappies}, {Crisps, Nappies}}

n = 2

$\mathbb{I}_2$ = {{Beer, Crisps}, {Beer, Nappies}, {Crisps, Nappies}}

$\mathcal{I}$ = {{Beer}, {Crisps}, {Nappies}, {Beer, Crisps}, {Beer, Nappies}, {Crisps, Nappies}}

H_3 = {{Beer, Crisps, Nappies}}

n = 3

$\mathbb{I}_3$ = {{Beer, Crisps, Nappies}}

$\mathcal{I}$ = {{Beer}, {Crisps}, {Nappies}, {Beer, Crisps}, {Beer, Nappies}, {Crisps, Nappies}, {Beer, Crisps, Nappies}}

H_4 = {}

n = 4

$\mathbb{I}_4$ = {}

Algorithm terminated

In[32]:=
```
popularSets = (# → support[#] &) /@ 𝓘;
SortBy[popularSets, #[[2]] &] // TableForm
```

Out[33]//TableForm=

{Beer, Crisps} → $\frac{1}{2}$

{Crisps, Nappies} → $\frac{1}{2}$

{Beer, Crisps, Nappies} → $\frac{1}{2}$

{Beer} → $\frac{3}{4}$

{Crisps} → $\frac{3}{4}$

{Nappies} → $\frac{3}{4}$

{Beer, Nappies} → $\frac{3}{4}$

Step 2

In[34]:=
```
rules = Flatten[
    ((* Iterate over each element in 𝓘 *)
      Module[{element, singleElements, singleElement, compl, konf, result = {}},
       element = #; (* {"Drucker","Hut","Schal"} *)

       Do[
        (* All i-th parts of the current element:
            i=1: {{"Drucker"},{"Hut"},{"Schal"}} and
            i=2: {{"Drucker","Hut"},{"Drucker","Schal"},{"Hut","Schal"}}
        *)
        singleElements = Subsets[element, {i}];

        ((* Iterate over each element in the subset,
           place each element on the right side *)
           singleElement = #; (* {"Drucker"} *)
           compl = Complement[element, singleElement]; (* {"Hut","Schal"} *)
           konf = confidence[compl, singleElement];
           (* {"Hut","Schal"} → {"Drucker"} *)
           AppendTo[result, {compl → singleElement, konf}];
          ) & /@ singleElements;

        , {i, 1, Length[element] - 1}];

       result
      ]
     ) & /@ Cases[𝓘, Y_ /; Length[Y] ≥ 2], 1] // N;
```

All rules:

In[35]:=
```
rules // TableForm
```

Out[35]//TableForm=

{Crisps} → {Beer}	0.666667
{Beer} → {Crisps}	0.666667
{Nappies} → {Beer}	1.
{Beer} → {Nappies}	1.
{Nappies} → {Crisps}	0.666667
{Crisps} → {Nappies}	0.666667
{Crisps, Nappies} → {Beer}	1.
{Beer, Nappies} → {Crisps}	0.666667
{Beer, Crisps} → {Nappies}	1.
{Nappies} → {Beer, Crisps}	0.666667
{Crisps} → {Beer, Nappies}	0.666667
{Beer} → {Crisps, Nappies}	0.666667

Only rules which fulfil the confidence constraint:

In[36]:=
```
Cases[
  SortBy[rules, #[[2]] &],
  Y_ /; Y[[2]] ≥ kmin
 ] // TableForm
```

Out[36]//TableForm=

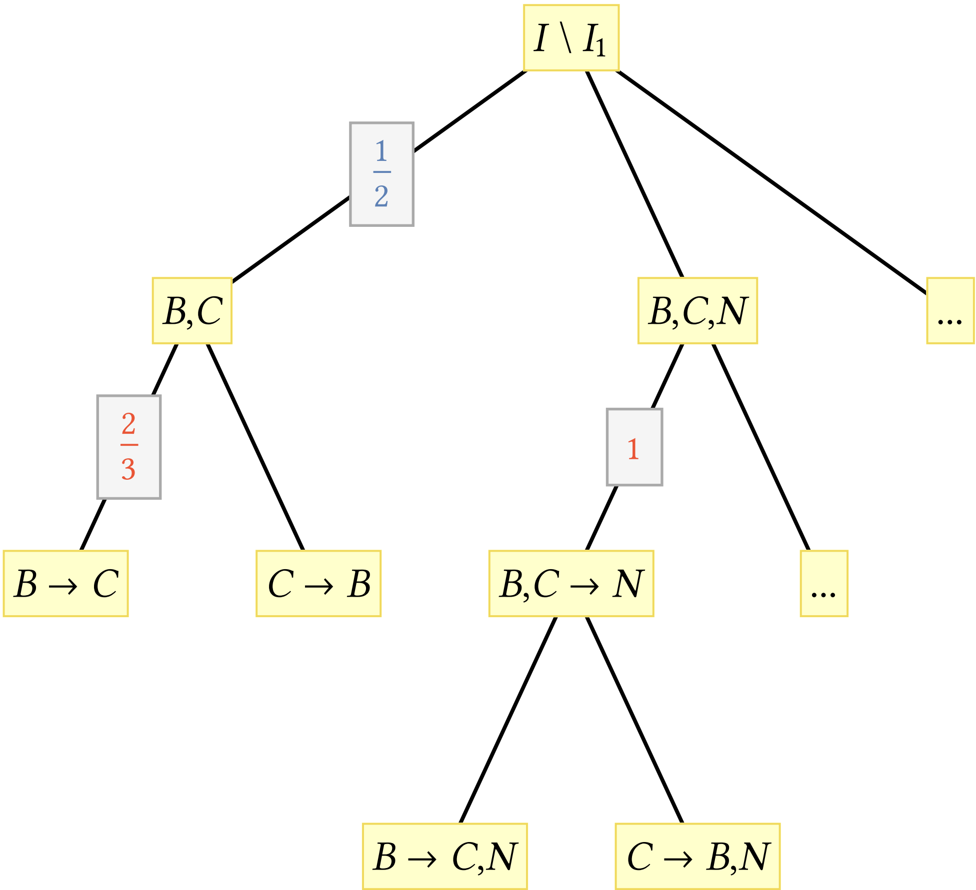
{Beer} → {Nappies}	1.
{Nappies} → {Beer}	1.
{Beer, Crisps} → {Nappies}	1.
{Crisps, Nappies} → {Beer}	1.

In[37]:=
```
reLabelF[verts_, labels_][tp_] :=
 tp /. (Framed[#, p__] :> Framed[#2, p] &@@@ Transpose[{verts, labels}])

labels = {
   Row[{it["I"], " \\ ", "I₁"}],
   Row[{it["B"], ",", it["C"]}],
   Row[{it["B"], ",", it["C"], ",", it["N"]}],
   "...",
   Row[{it["B"], " → ", it["C"]}],
   Row[{it["C"], " → ", it["B"]}],
   Row[{it["B"], ",", it["C"], " → ", it["N"]}],
   "...",
   Row[{it["B"], " → ", it["C"], ",", it["N"]}],
   Row[{it["C"], " → ", it["B"], ",", it["N"]}]
  };

TreePlot[{
   {1 → 2, 1/2},
   1 → 3,
   1 → 4,
   {2 → 5, 2/3},
   2 → 6,
   {3 → 7, 1},
   3 → 8,
   7 → 9,
   7 → 10
  },
  VertexLabeling → True,
  ImageSize → Large,
  EdgeRenderingFunction →
   ({Thick, Line[#1], If[#3 === None, {}, Text[Panel@StandardForm@Style[#3,
          {  ■   #3 == 1/2
          {  ■   True        , FontFamily → "Libertinus Serif", FontSize → 18], Mean@#1]]} &)
 ] // reLabelF[Range[10], Style[#, 22, FontFamily → "Libertinus Serif"] & /@ labels]
```

Out[39]=



Supplementary questions

- We can see from the transaction table that the minimum confidence is $\frac{2}{3}$ for possible rules of I (this is also confirmed by the results above). To get a small value for the confidence, we need the numerator to be as small as possible and the denominator to be as large as possible, i.e.

$$\mathrm{conf}(X \to Y) = \frac{\mathrm{support}(X \cup Y)}{\mathrm{support}(X)} \Rightarrow \frac{\text{as small as possible}}{\text{as large as possible}} \Rightarrow \frac{0.4}{0.6} = \frac{2}{3}$$

  - $\mathrm{support}(X \cup Y) = \frac{2}{5}$ because in all combinations of $\{B, N, C\}$ at least two transactions remain.
  - $\mathrm{support}(X) = \frac{3}{5}$ because each item is not bought more than three times (X must be a single-item set since more items can only decrease the support).
- The rules $C \to T$ and $T \to C$ have both a confidence value of 0 since the two items do not co-occur.