Catalan Numbers

Charlene Chu - October 6, 2017

Homework: Figure out how Catalan numbers is related to binomial coefficients.

Project: Construct *Poster[n, a list of methods]* to draw the poster in Mathematica.

Triangulations

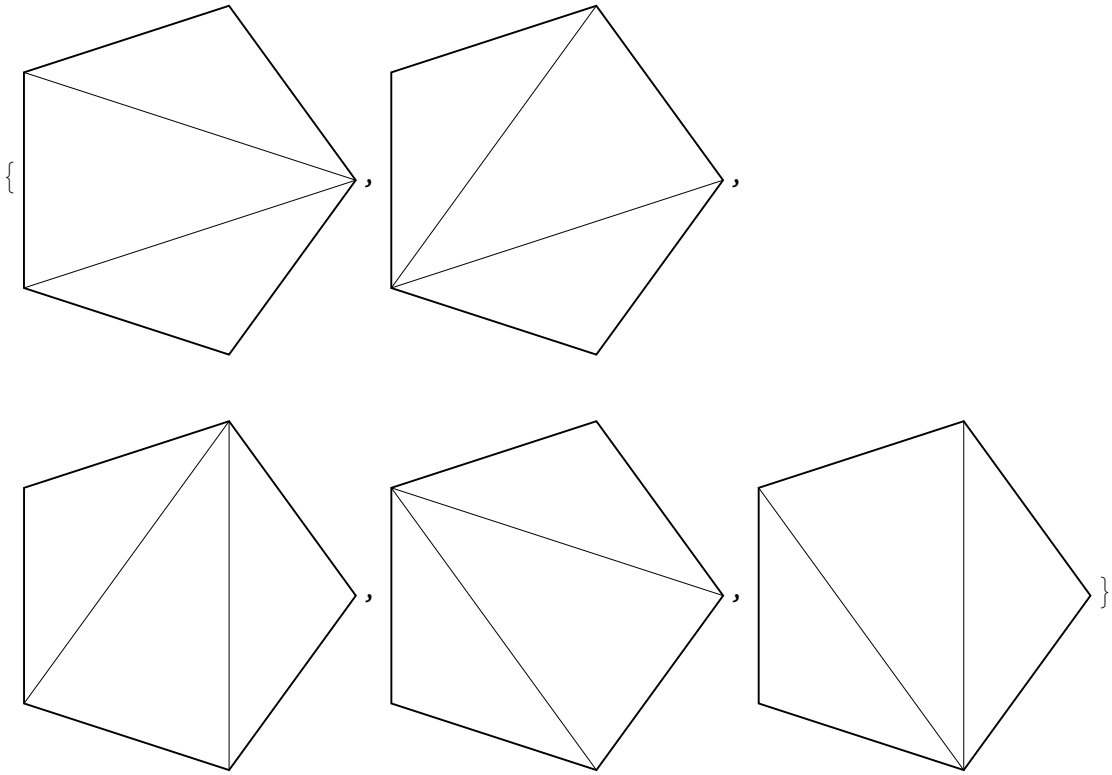
Triangulations[n] returns a list of triangulations of an (n+2)-gon.

```
(* create triangulations, from lecture *)
ts[n_Integer] := ts[Range[0, n + 1]];
ts[{_, _}] = {ds[]};
ts[vs_List] := Module[{l, r, k, t1, t2, tds},
   Union @@ Table[
     l = ts[Prepend[vs[[k ;;]], vs[[1]]]];
     r = ts[vs[[2 ;; k]]];
     Flatten[Table[
       tds = Join[t1, t2];
       If[k > 3, AppendTo[tds, d[vs[[2]], vs[[k]]]]];
       If[k < Length[vs], AppendTo[tds, d[vs[[1]], vs[[k]]]]];
       tds,
       {t1, l}, {t2, r}
      ]],
     {k, 3, Length[vs]}
    ]
  ];
```

```
(* create diagonals, from lecture *)
tsList[n_Integer] :=
 ts[n] /. ds[ls___] → Graphics[{ls}] /. d[i_, j_] :> Line[{i, j}] /.
  j_Integer :→ {Cos[2 π j / (n + 2)], Sin[2 π j / (n + 2)]}
```

```
(* draw diagonals together with the (n+2)-gon *)
triangulations[n_Integer] := Table[Graphics[Join[
    tsList[n][[m, 1]],
    {EdgeForm[Thickness[Medium]], Transparent, RegularPolygon[{1, 0}, n + 2]}
   ]], {m, 1, Length[tsList[n]]}]
```

```
(* example *)
triangulations[3]
```



Non-associative words

sentences[n] returns a list of “sentences” with n+1 words.

```
(* helper function *)
multiplyWords[A_List, B_List] := Module[{result, temp1, temp2},
  temp1 := {A, B};
  temp2 := B /. Intersection[Flatten[A], B][[1]] :→ A;
  If[ Intersection[Flatten[A], B] ≠ {}, temp2, temp1]
 ]
```

```
(* create nested list to simulate the multiplication *)
sentencesList[n_] := Module[{edges, newSentence, nextWord, missingWord, tempSentences},
  edges = ts[n] /. d[i_, j_] :> {i + 1, j} /. ds[ls___] :> {ls};
  tempSentences  = Table[
    newSentence = words[[1]];
    If[
     Length[words] > 1,
     nextWord = words[[2]];
     For[i = 1, i < Length[words] - 1, ++i,
      {newSentence, nextWord} = {multiplyWords[newSentence, nextWord], words[[i + 2]]}];
     newSentence = multiplyWords[newSentence, nextWord],
     (* else, do nothing*)
    ];
    missingWord = Complement[Range[n + 1], Flatten[newSentence]] ;
    If[missingWord == {}, newSentence,
     missingWord = missingWord[[1]];
     If[ missingWord > Max[Flatten[newSentence]], newSentence =
       {newSentence, missingWord}, newSentence = {missingWord, newSentence}]
    ],
    {words, Table[SortBy[itm, Differences], {itm, edges}]}
   ]
 ]
```

```
(* incomplete function. if the list is {{x1,x2},x3}, then we want (x1x2)x3 *)
sentences[n_] := Module[{tempStr},
  Table[
   (* tempStr = word/. j_Integer :> xj;*)
   tempStr = ToString[word];
   tempStr = StringReplace[tempStr, {"{" → "(", "}" → ")", "," → ""}],
   {word, sentencesList[n]}
  ]
 ]
```

```
(* example *)
sentences[3]
```

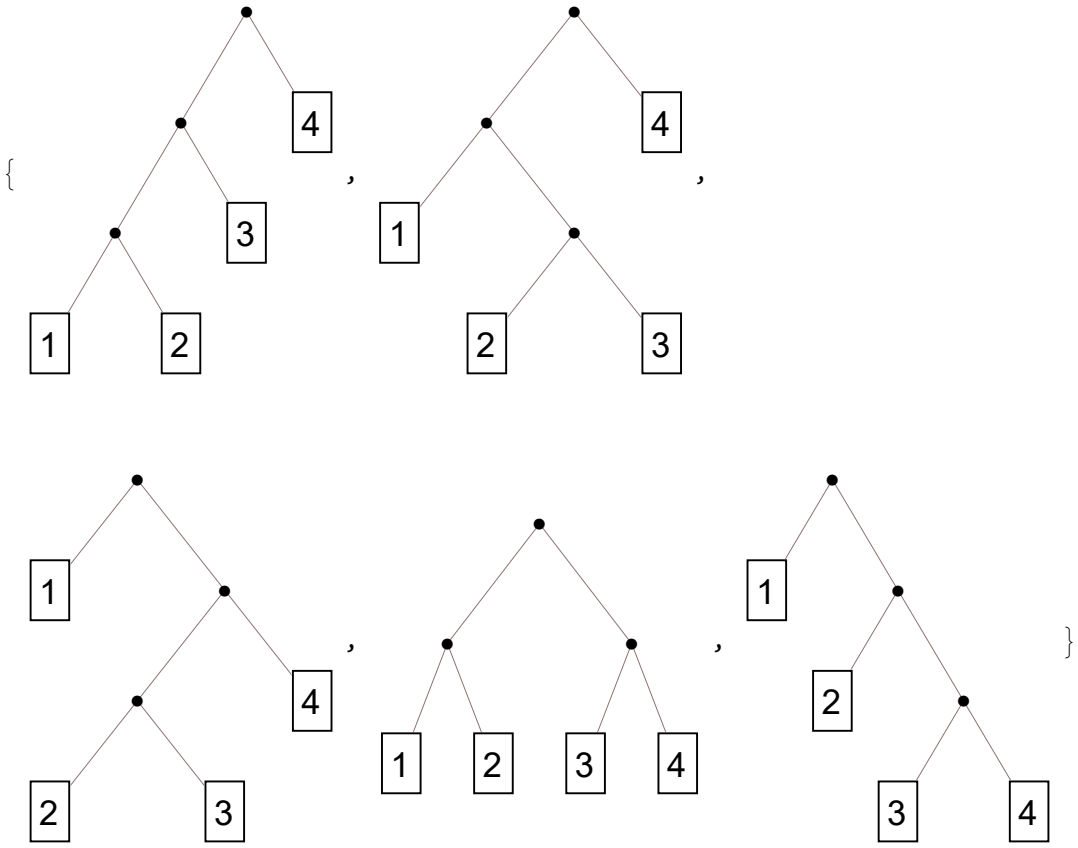
```
{(((1 2) 3) 4), ((1 (2 3)) 4), (1 ((2 3) 4)), ((1 2) (3 4)), (1 (2 (3 4)))}
```

Trees

trees[n] returns a tree that corresponds to triangulations[n] andsentences[n].

```
trees[n_] := Table[TreeForm[itm,
   (* Source: https://mathematica.stackexchange.com/questions/81247/how-
     to-label-only-leaves-in-treeform *)
   VertexRenderingFunction → (If[#2 === List, Inset[Text["●"], #],
       Inset[Framed[Text[Style[#2, 18]], Background → White], #]] &)
  ], {itm, sentencesList[n]}]
```

```
(* example *)
trees[3]
```



Final Product

Poster[n,methods] returns a poster with the the triangulations, non-associative words, and trees, for n>1.

```
Poster[n_Integer, methods_List] := Module[{m, l, p},
  (* methods is list of three elements: 0 means do not include this metho,
  1 means include this method *)
  m = {triangulations[n], sentences[n], trees[n]};
  l = Length[m[[1]]];
  p = Position[methods, 1];
  Grid[
   Table[
    m[[i, j]][[1]],
    {i, p},
    {j, l}],
   Frame → All]
 ]
```

```
(* example *)
Poster[3, {1, 1, 1}]
```