Pensieve header: This is the index file for the PPSA project.

Make

```
Make::usage =
  "Make[targets, sources, Hold[action]] makes a target, or a list of targets, given sources,
    or a list of sources, in the style of the unix 'make' command.";
Make[target_String, sources_, action_Hold] := Make[Evaluate@{target}, sources, action];
Make[targets_, source_String, action__Hold] := Make[targets, Evaluate@{source}, action];
Make[targets_List, sources_List, action_Hold] := Module[{},
   If[
    (And @@ ((FileType[#] =!= None) & /@ sources)) &&
     Or[
      Or @@ ((FileType[#] === None) & /@ targets),
      Min[AbsoluteTime[FileDate[#]] & /@ targets] < Max[AbsoluteTime[FileDate[#]] & /@ sources]
     ],
    Print["Making ", targets, " ..."];
    ReleaseHold[action]
   ]
  ];
```

WordCloud

```
sources = {"PPSA.tex", "abstract.tex", "intro.tex", "odds.tex", "refs.tex"};
target = "WordCloud.png";
```

```
(*ReadFile[fname_] := Module[{lines},
  lines=DeleteCases[StringSplit[ReadString[fname], "\n"], ""];
  lines=First[StringSplit[#, "%", All]]& /@ lines;
  StringJoin@@lines
 ];
MakeWC[]:= Module[{words,dict},
  words=Flatten[TextWords[ReadFile[#]]& /@ sources];
  dict=Complement[
    DeleteStopwords[DictionaryLookup[]],
    {"begin", "end","left","right", "equation", "item","em"}
   ];
  words =Select[words, MemberQ[dict,#]&];
  WordCloud[words, ImageSize→400]
 ]*)
```

```
MakeWC[opts___] := Module[{words, words1, dict, T, dict1},
  words = ToLowerCase@DeleteStopwords@Flatten[
      StringSplit[TextWords[ReadString[#]], "-"] & /@ sources
     ];
  dict = Complement[
    Union[ToLowerCase@DictionaryLookup[], StringSplit[
      "aarhus abelian acknowledgements adjoint adjoints albert alekseev alexander antipode anton
        archibald artin arxiv associator associators bardakov basepoint behaviour berceanu
        bialgebra bialgebras bijection borromean brendle brochier cablings centres chern chu
        claspers coadjoint cocommutative cocycle coface cofactor colour coloured colourful
        colourings colours combinatorially combinatorics componentwise conjecturally crans
        dancso det diffeomorphism drinfeld dror duflo enriquez equivariant etingof exp
        exponentiate fenn fibre flavours formulae framings functionals functor functorial
        functoriality functors furusho gluings goussarov grothendieck grouplike habiro
        halacheva harinck hatcher haviv homfly homomorphic homomorphicity homonymous
        homotopic homotopies hopf ihx injective isometries isotopies isotopy jacobian
        kamnitzer kanenobu karene kashiwara kauffman kazhdan  kishino kneissler knottings
        kohno kontsevich kricker kuperberg kurlin lescop leung lieberum linearization
        linearizations loday mccool meilhan meinrenken metrized milnor moded moding
        modulo multicategory multinary multiplicatively naot natan ohtsuki operad
        overcrossing overcrossings papadima parametrizing parentehesized parentetization
        parenthesization parenthesizations parenthetization perturbative planarity postfix
        preprint projectivization projectivizations proven quadrivalent quandle quandles
        reassociate reidemeister reutenauer rimanyi rolfsen roukema saito sanderson
        satoh sder selflinking semidirect semivirtual shima simons sinh skeleta skype
        subalgebra subalgebras subring surjection surjections surjective symmetrized tder
        teichmuller thurston torossian tr trivalence trivolution unbraided undercrossing
        undercrossings unfavourably unforbidden unignoring unipotent unital unitarity
        unitrivalent unknot unoriented usb valent vassiliev vergne verma versa vertices
        virtuals voldemort warmup watanabe wirings wirtinger wko zhang zsuzsanna zsuzsi"
     ]],
    StringSplit["ac aft alpha begin beta bullet cali dj em end
       equation eta fa ill left minus natan plus red ref rh right rs ts tv ty xi"]
   ];
  dict1 = Dispatch[(# → T[#]) & /@ dict];
  words1 = Cases[words /. dict1, T[w_] :> w, {1}];
  WordCloud[words1, opts]
 ]
```

```
MakeWC[ImageSize → 400]
```


```
Make[target, Join[sources, {"index.nb"}], Hold[
  Export[target, MakeWC[ImageSize → 420]];
  MakeThumb@target;
 ]]
```

Output

```
{
 "TitleNotes" ->
  StringJoin["<div style=\"clear: right; float: right; padding: 8px; width: 400px;\"><img
     width=400px src=WordCloud.png></div>This is the construction / computation page
     for my joint paper with <a class=external href=\"http://www.rolandvdv.nl/\">Roland
     van der Veen</a>, <b>Poly-Time Knot Polynomials Via Solvable
     Approximations</b> (<a href=PPSA.pdf>PDF here</a>). <p><b>Abstract.</b> ",
   StringReplace[ReadString["abstract.tex"], {
     "~" → " ",
     "\\cite[1928]{Alexander:TopologicalInvariants}" → "[<a href=Al>Al</a>, 1928]",
     "\\cite{Rozansky:Flat1,Rozansky:Burau,Rozansky:U1RCC}" →
      "[<a href=Ro1>Ro1</a>, <a href=Ro2>Ro2</a>, <a href=Ro3>Ro3</a>]",
     "\\cite{Overbay:Thesis}" → "[<a href=Ov>Ov</a>]"
    }],
   "<p><b>Quick Links.</b> <a href=Ov>Ov</a>.",
   "\n"
  ]
}
```

```
{TitleNotes → <div style="clear: right; float: right; padding: 8px; width: 400px;"><img width=400px
    src=WordCloud.png></div>This is the construction / computation page for my joint paper with
    <a class=external href="http://www.rolandvdv.nl/">Roland van der Veen</a>, <b>Poly-Time Knot
    Polynomials Via Solvable Approximations</b> (<a href=PPSA.pdf>PDF here</a>). <p><b>Abstract.</b>
    Following Rozansky [<a href=Ro1>Ro1</a>, <a href=Ro2>Ro2</a>, <a href=Ro3>Ro3</a>]
and Overbay [<a href=Ov>Ov</a>], we construct
the first poly-time-computable knot polynomials since
Alexander's [<a href=Al>Al</a>, 1928]. We use
some new commutator-calculus techniques and a family of Lie algebra
$sl_2^{\leq k}$ which are solvable yet at the same time they make progressively
better approximations of the simple Lie algebra $sl_2$. The resulting
invariants are the strongest genuinely-computable knot invariants
presently available and they seem to contain information about some
classical topologically-defined knot invariants.
<p><b>Quick Links.</b> <a href=Ov>Ov</a>.
}
```