

Pensieve header: The Splash Program of Figure 1.1, computing \$z_1\$.

```
SetDirectory["C:\\drorbn\\AcademicPensieve\\Projects\\PPSA\\PPSA-170103"];
```

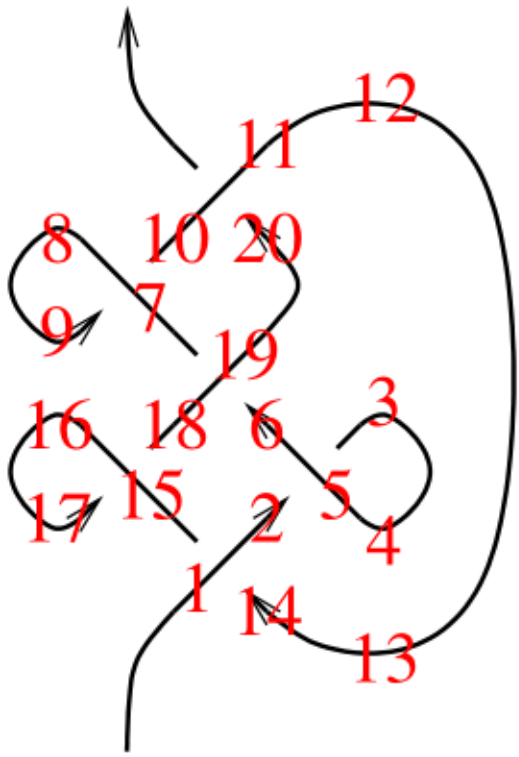
Splash Demo

Program

```
DPx→Dα, y→Dβ[P_][f_] := (* means P[∂α, ∂β][f] *)
Total[CoefficientRules[P, {x, y}] /. ({m_, n_} → c_) ⇒ c D[f, {α, m}, {β, n}]];
CF[E[ω, L_, Q_, P_] := Expand /@ Together /@
E[ω / . b ⇒ Log[t], L, Q / . b ⇒ Log[t], P / . b ⇒ Log[t]];
E /: E[ω1_, L1_, Q1_, P1_] E[ω2_, L2_, Q2_, P2_] := CF@E[ω1 ω2, L1 + L2, ω2 Q1 + ω1 Q2, ω24 P1 + ω14 P2];
Nui, cj→k-[E[ω, L_, Q_, P_]] := With[{q = e-y β uk + γ ck}, CF[
E[ω, γ ck + (L / . cj → θ), ω e-y β uk + (Q / . ui → θ), e-q DPcj→Dγ, ui→Dβ[P][eq]] /. {γ → ∂cjL, β → ω-1 ∂uiQ}]];
Nwi, cj→k-[E[ω, L_, Q_, P_]] := With[{q = ey α wk + γ ck}, CF[
E[ω, γ ck + (L / . cj → θ), ω ey α wk + (Q / . wi → θ), e-q DPcj→Dγ, wi→Dα[P][eq]] /. {γ → ∂cjL, α → ω-1 ∂wiQ}]];
Nwi, uj→k-[E[ω, L_, Q_, P_]] := With[{q = (1 - t) μ-1 α β + μ-1 β uk + μ-1 δ uk wk + μ-1 α wk}, CF[
E[μ ω, L, μ ω q + μ (Q / . wi | uj → θ), μ4 e-q DPwi→Dα, uj→Dβ[P][eq] + ω4 Δ[k]] /. μ → 1 + (t - 1) δ / .
{α → ω-1 (∂wiQ / . uj → θ), β → ω-1 (∂ujQ / . wi → θ), δ → ω-1 ∂wi, ujQ}]];
mi, j→k-[Z_] := Module[{x, y, z},
Z // Nui, cj→x // Nwi, uj→y // ReplaceAll[{cx|y → cx, wj → wy}] // Nui, cx→x // ReplaceAll[z-i|j|xy → zk] // CF]
```

Data

```
Δ[k_] := (1 - t) (α2 β2 + 4 α β δ μ + 2 δ2 μ2) / 2 + 2 μ2 (α β + δ μ) ck - β (2 μ - 1) (α β + 2 δ μ) uk +
2 β δ μ2 ck uk - β2 δ (3 μ - 1) uk2 / 2 + α (α β + 2 δ μ) wk + 2 α δ μ2 ck wk - 2 (t - 1) δ2 (α β + δ μ) uk wk +
2 δ2 μ2 ck uk wk - β δ2 (2 μ - 1) uk2 wk + α2 δ (1 + μ) wk2 / 2 + α δ2 uk wk2 - (t - 1) δ4 uk2 wk2 / 2;
Ri, j-+ := E[1, bi cj, ui wj, -ci (t - 1)2 / 2 - ci2 (t - 1)2 / 2 + ci cj (t2 - t - 2) / 2 - cj ui wi / 2 + ci (1 - t) ui wi -
ui2 wi2 / 2 + ui wj + cj t ui wj / 2 + ci (t - 2) t ui wj + ci (1 + t) uj wj / 2 + (t - 1) ui2 wi wj - (t - 2) t ui2 wj2 / 2];
Ri, j-- := E[1, -bi cj, -t-1 ui wj, ci (t - 1)2 / 2 + ci2 (t - 1)2 / 2 + ci cj (2 + t - t2) / 2 + cj ui wi / 2 +
ci (t - 1) ui wi + ui2 wi2 / 2 + (1 - t-1) ui wj / 2 + ci (2 t - 5 + 3 t-1) ui wj / 2 + cj (t-1 + 1 - t) ui wj / 2 -
ci (t + 1) uj wj / 2 + (2 - 3 t-1) ui2 wi wj / 2 + (1 + 2 t-2 - 3 t-1) ui2 wj2 / 2 - t-1 (1 + t) ui uj wj2 / 2];
uri := E[t-1/4, 0, 0, ci t / 4 + ui wi / 8];
nri := E[t1/4, 0, 0, -ci t3 / 4 - t2 ui wi / 8];
uli := E[t1/4, 0, 0, ci t (4 + t) / 4 - t2 ui wi / 8];
nli := E[t-1/4, 0, 0, -ci (1 + 4 t-1) / 4 + ui wi / 8];
```



Run

```
z = R1,14+ R5,2- nr3 ul4 R19,6+ R7,10- nl8 ur9 R11,20+ nr12 ul13 R15,18- nl16 ur17;
(Do[z = z // m1,k→1, {k, 2, 20}]; z = z /. a-1 → a)
```

Run

$$\mathbb{E} \left[-1 + \frac{1}{t} + t, 0, 0, -16 + \frac{9c}{2} - \frac{2c}{t^4} + \frac{1}{t^3} + \frac{11c}{2t^3} - \frac{4}{t^2} - \frac{8c}{t^2} + \frac{10}{t} + \frac{4c}{t} + 18t - 10ct - 14t^2 + 8ct^2 + 7t^3 - \frac{3ct^3}{2} - 2t^4 - 2ct^4 + 2ct^5 - \frac{ct^6}{2} - 4uw + \frac{2uw}{t^4} - \frac{7uw}{2t^3} + \frac{9uw}{2t^2} + \frac{uw}{2t} + 6t uw - 2t^2 uw - \frac{1}{2} t^3 uw + \frac{3}{2} t^4 uw - \frac{1}{2} t^5 uw \right]$$

Exporting the above as PDF files

The below is adapted from pensieve://2016-04/GaussGassner/GaussGassnerDemo.nb.

```
ConditionalExport[fname_String, rest___] := Module[{temp, exists},
  temp = "ConditionalExportTemporary" <> "." <> FileExtension[fname];
  exists = FileExistsQ[fname];
  Export[temp, rest];
  If[exists && FileByteCount[fname] === FileByteCount[temp],
    DeleteFile[temp],
    (* else *) Print["Exporting " <> fname <> "..."];
    If[exists, DeleteFile[fname]];
    RenameFile[temp, fname]
  ];
  fname
]
```

```
SetOptions[$FrontEndSession, PrintingStyleEnvironment → "Working"];
TagProperties[_] := {};
TagProperties["ct-def"] = {PageWidth → 5/0.66};
Options[CellExport] = {
  PageWidth → 3.2/0.66 * 4.2/3.8, CellFilter → Identity, ExportDirectory → "Snips",
  ExportBaseFilename → Automatic, ExportFormat → ".pdf", ExportOptions → {}, Split → False
};
```

```

CellExport[tag_String, opts__Rule] := CellExport[
  NotebookGet[EvaluationNotebook[]],
  tag, opts
];
CellExport[nb_Notebook, tag_String] := CellExport[nb, tag, TagProperties[tag]];
CellExport[nb_Notebook, tag_String, OptionsPattern[]] := Module[
  {cells, cell, filename, format},
  filename = FileNameJoin[{
    OptionValue[ExportDirectory] /. Automatic -> Directory[],
    OptionValue[ExportBaseFilename] /. Automatic -> tag
  }];
  format = OptionValue[ExportFormat];
  cells = OptionValue[CellFilter][Cases[
    nb, c_Cell /; FreeQ[List@@c, Cell] && !FreeQ[c, CellTags -> tag],
    Infinity
  ]];
  If[!OptionValue[Split],
    If[Length[cells] ≥ 1,
      If[Length[cells] == 1,
        cells = Join[First[cells],
          Cell[PageWidth -> 1.2 × 72 OptionValue[PageWidth], Background -> {White, Opacity[0]}]],
        cells = Cell[CellGroup[cells], PageWidth -> 72 OptionValue[PageWidth]]
      ];
      ConditionalExport[
        filename <> format, cells,
        ImageResolution -> 300,
        OptionValue[ExportOptions]
      ]
    ],
    k = 0;
    Table[
      ++k;
      ConditionalExport[
        filename <> "-" <> ToString[k] <> format,
        Append[cell, PageWidth -> 72 OptionValue[PageWidth]],
        ImageResolution -> 300,
        OptionValue[ExportOptions]
      ],
      {cell, cells}
    ]
  ];
];

```

```

ExportCells := (
  nb = NotebookGet[EvaluationNotebook[]];
  tags = Cases[nb, (CellTags -> tag_String) => tag, Infinity] // Union;
  Print[tags];
  CellExport /@ tags;
  Print["Done."]
);

```

ExportCells

{Data, Program, Run}

Exporting Snips\Data.pdf...

Exporting Snips\Program.pdf...

Exporting Snips\Run.pdf...

Done.