

Pensieve header: OneCo, the local picture. Bug-fixing branch for use in acirc.nb - ca renamed to CA.

```
Print["In the U(T)⊗U(H) conventions. Internal use symbols: ", {rr, pp}]
```

```
In the U(T)⊗U(H) conventions. Internal use symbols: {rr, pp}
```

```
SetDirectory["C:\\drorbn\\AcademicPensieve\\Projects\\OneCo-1604"]
```

```
C:\\drorbn\\AcademicPensieve\\Projects\\OneCo-1604
```

Generalities

Generalities

```
Simp[ε_] := Simplify[ε];
CF[ε_] := ε /. λ_β | λ_a | λ_δβ | λ_δa | λ_δaa => MapAt[Simp, λ, 1];
AutoCollecting[λ_] := (λ /: λ[0, ___] = 0;
  λ /: λ[f_, r___] + λ[g_, r___] := λ[Simp[f+g], r];
  λ /: g_*λ[f_, r___] := λ[Simp[g f], r]);
AutoCollecting /@ {β, a, δβ, δa, δaa};
UU /: UU[x_] + UU[y_] := UU[x+y];
UU /: a_*UU[x_] := UU[Expand[a x]];
UU /: D[u_UU, vs___] :=
  CF[u /. λ_β | λ_a | λ_δβ | λ_δa | λ_δaa => MapAt[D[#, vs] &, λ, 1]];
b_φ = 1;
(* Phase out: *) CA[f_, i_, j_, k_] := δaa[f, φ, i, j, k];
γa[f_, j_, k_, l_, m_] := δaa[f, j, k, l, m] - CA[b_j f, k, l, m];
hb[f_, i_, j_, k_] := a[b_j f, i, k] - a[b_i f, j, k] (*=f[a_ik, a_jk]*);
δhb[f_, i_, j_, k_] := δa[b_j f, i, k] - δa[b_i f, j, k];
δtb[f_, x_, j_, l_] := CA[f, l, x, j] - CA[f, j, x, l];
UU /: Coefficient[u_UU, λ_[js___]] := Total[Cases[u, λ[f_, js] => f, ∞]];
Kδ /: Kδ_{is} := KroneckerDelta[1, Length[Union[{is}]]];
```

δaa relations

delta-aa

```
i_ ≤ j_ := OrderedQ[{i, j}]; i_ < j_ := !OrderedQ[{j, i}];
CF[UU[ε_]] := UU[CF[ε /. δaa[f_, i_, j_, k_, l_] => Which[
  k === φ, δaa[f, φ, l, i, j] + Kδ_{j1} δhb[f, i, φ, j],
  (i === φ) ∨ (i ≤ k ∧ j ≤ l), δaa[f, i, j, k, l],
  k < i ∧ j < l, δaa[f, k, j, i, l] + CA[-f b_i, l, k, j] +
  CA[f b_i, j, k, l] + CA[-f b_k, j, i, l] + CA[f b_k, l, i, j],
  k < i ∧ j === l, δa[-f b_i, k, j] + δa[f b_k, i, j] + δaa[f, k, j, i, j],
  i ≤ k ∧ l < j, δaa[f, i, l, k, j] + CA[-f b_i, l, k, j] +
  CA[f b_i, j, k, l] + CA[-f b_k, j, i, l] + CA[f b_k, l, i, j],
  k < i ∧ l < j, δaa[f, k, l, i, j]
]]];
```

Bases

```

UUBasis[T_List, H_List, f_] := Module[
  {ff, n = 0, h, t, h1, h2},
  ff := f_{++n} @@ Table[b_t, {t, T}];
  CF /@ UU /@ Flatten@{
     $\beta$ [ff],  $\delta\beta$ [ff],
    Table[a[ff, t, h], {t, T}, {h, H}],
    Table[ $\delta a$ [ff, t, h], {t, T  $\cup$  { $\mathfrak{c}$ }}, {h, H}],
    Table[CA[ff, h1, t, h2], {t, T}, {h1, H}, {h2, H}],
    Table[ $\delta aa$ [ff, T[[i]], H[[j]], T[[k]], H[[l]],
      {k, Length@T}, {i, k}, {l, Length@H}, {j, l}]
  ] /. 1_[___]  $\rightarrow$  1
];

UUBasis[S_List, f_] := UUBasis[S, S, f];
UUBasis[n_Integer, m_Integer, f_] := UUBasis[Range@n, Range@m, f];
UUBasis[n_Integer, f_] := UUBasis[Range@n, f];
hooBasis[S_List, f_] := Module[{n = 0},
  Select[UUBasis[S, S  $\cup$  {hoo}, f], Count[#, hoo, {2}] == 1 &] /. f_[bs___]  $\Rightarrow$  f_{++n}[bs]
];

```

NonCommutativeMultiply

NonCommutativeMultiply

```

UU /: UU[L_] ** UU[R_] := CF@UU[Expand[Distribute[pp[L, R]] /. {
  pp[0, _]  $\rightarrow$  0, pp[_ , 0]  $\rightarrow$  0,
  pp[ $\beta$ [f_],  $\lambda_\beta$  |  $\lambda_a$  |  $\lambda_{\delta\beta}$  |  $\lambda_{\delta a}$  |  $\lambda_{\delta aa}$ ]  $\Rightarrow$  MapAt[f # &,  $\lambda$ , 1],
  pp[_  $\delta\beta$  | _  $\delta a$  | _  $\delta aa$ , _  $\delta\beta$  | _  $\delta a$  | _  $\delta aa$ ]  $\rightarrow$  0,
  pp[ $\delta a$ [f_, i_, j_], a[g_, k_, l_]]  $\Rightarrow$ 
     $\delta aa$ [fg, i, j, k, l] -  $K\delta_{jk}\delta hb$ [fg, i, j, l],
  pp[ $\delta\beta$ [f_],  $\beta$ [g_]]  $\Rightarrow$   $\delta\beta$ [fg],
  pp[ $\delta\beta$ [f_], a[g_, i_, j_]]  $\Rightarrow$   $\delta a$ [fg, i, j]
}]];

```

tm, hm, hts, dm

tm-def

```
UU[ε_] // tm[x_, y_, z_] := (rr = Replace[x | y → z];
  CF[UU[Expand[ε /. {
    a[f_, x, j_] := a[f, z, j] + δhb[∂byf, z, ϕ, j],
    a[f_, y, j_] := a[f, z, j],
    δa[f_, x | y, j_] := δa[f, z, j],
    δaa[f_, i_, j_, k_, l_] := δaa[f, rr@i, j, rr@k, l]
  } /. bx|y → bz]]]);
```

hm-def

```
UU[ε_] // hm[x_, y_, z_] := (rr = Replace[x | y → z];
  CF[UU[Expand[ε /. {
    a[f_, i_, x | y] := a[f, i, z],
    δa[f_, i_, x | y] := δa[f, i, z],
    δaa[f_, i_, y, k_, x] := δaa[f, k, z, i, z],
    δaa[f_, i_, j_, k_, l_] := δaa[f, i, rr@j, k, rr@l]
  }]]]);
```

hts-def

```
UU[ε_] // hts[y_, x_] := CF[UU[Expand[ε /. {
  a[f_, i_, j_] := a[f, i, j] -
    Kδjy δhb[∂bxf, i, ϕ, y] - Kδix Kδjy (β[f bx] - δa[f, ϕ, y] - δβ[bx ∂bxf]),
  δa[f_, x, y] := δa[f, x, y] - δβ[f bx],
  δaa[f_, i_, j_, k_, l_] := δaa[f, i, j, k, l] +
    Kδix Kδjy δa[-bx f, k, l] + Kδix Kδly (δa[bk f, x, j] - δa[bx f, k, j]) +
    Kδkx Kδjy (δa[bi f, x, l] - δa[bx f, i, l]) + Kδkx Kδly δa[-bx f, i, j] -
    Kδix Kδjly δβ[bx bk f] + 2 Kδxik Kδyjl δβ[bx bx f]
  }]]];
```

dm-def

```
dm[x_, y_, z_][ε_] := ε // hts[x, y] // tm[x, y, z] // hm[x, y, z]
```

$t\sigma, h\sigma, d\sigma$ on $\{\beta, a, \delta\beta, \delta a, \delta aa\}$

sigma-def

```

tσ[x_List, y_List][ε_] := (rr = Replace[Thread[x → y]]];
  CF[ε /. b_i → b_rr@i /. {
    a[f_, i_, j_] → a[f, rr@i, j],
    δa[f_, i_, j_] → δa[f, rr@i, j],
    δaa[f_, i_, j_, k_, l_] → δaa[f, rr@i, j, rr@k, l]
  }]);
tσ[x_, y_][ε_] := tσ[{x}, {y}][ε];
hσ[x_List, y_List][ε_] := (rr = Replace[Thread[x → y]]];
  CF[ε /. {
    a[f_, i_, j_] → a[f, i, rr@j],
    δa[f_, i_, j_] → δa[f, i, rr@j],
    δaa[f_, i_, j_, k_, l_] → δaa[f, i, rr@j, k, rr@l]
  }]);
hσ[x_, y_][ε_] := hσ[{x}, {y}][ε];
dσ[x_, y_][ε_] := ε // tσ[x, y] // hσ[x, y];

```

tb, hb, thb, htb, db, bb on $\{\beta, a, \delta\beta, \delta a, \delta aa\}$

tb-def

```

tb[x_][UU[L_], UU[R_]] := CF[UU[Expand[Distribute[pp[L, R]] /. {
  pp[0, _] → 0, pp[_ , 0] → 0,
  pp[_β | _δβ | _δa | _δaa, _β | _δβ | _δa | _δaa] → 0,
  pp[u_β | u_δβ | u_δa | u_δaa, v_a] → -pp[v, u]
} /. {
  pp[a[f_, x, j_], u_] → (u /. {
    β[g_] → δhb[f ∂bxg, x, ϕ, j],
    a[g_, k_, l_] →
      γa[f ∂bxg, x, j, k, l] + Kδxx (-γa[g ∂bxf, k, l, x, j] + δtb[fg, x, j, l]),
    _ → 0
  }],
  pp[a[f_, j_, k_], a[g_, x, l_]] /; j != x → -γa[g ∂bxf, x, l, j, k],
  pp[_ , _] → 0
}]]];

```

hb-def

```

hb[y_][UU[L_], UU[R_]] := CF[UU[Expand[Distribute[pp[L, R]] /. {
  pp[0, _] → 0, pp[_ , 0] → 0,
  pp[_β | _δβ, _] → 0,
  pp[_ , _β | _δβ] → 0,
  pp[_δa | _δaa, _δa | _δaa] → 0,
  pp[u_δa | u_δaa, v_a] := -pp[v, u]
} /. {
  pp[a[f_, i_, y], u_] := (u /. {
    a[g_, j_, k_] := Kδyk hb[f g, i, j, k],
    δa[g_, j_, k_] := Kδyk δhb[f g, i, j, k],
    δaa[g_, j_, k_, l_, m_] := Kδyk (δaa[bj f g, i, y, l, m] - δaa[bi f g, j, k,
      l, m]) + Kδym (δaa[bi f g, j, k, i, y] - δaa[bi f g, j, k, l, m])
  }),
  _pp → 0
}]]];

```

thb-def

```

thb[x_, y_][UU[L_], UU[R_]] := CF[UU[Expand[Distribute[pp[L, R]] /. {
  pp[0, _] → 0, pp[_ , 0] → 0,
  pp[_β | _δβ | _δa | _δaa, _β | _δβ | _δa | _δaa] → 0,
  pp[_a, _β | _δβ] → 0,
  pp[β[f_], a[g_, i_, j_]] := Kδyj δhb[g ∂bx f, i, ϕ, y],
  pp[a[f_, i_, j_], a[g_, k_, l_]] := Kδyl (
    γa[g ∂bx f, k, l, i, j] + Kδxi (
      δhb[-bk g ∂bx f, i, ϕ, j] + δa[bk g ∂bx f, i, j] - δa[bi g ∂bx f, k, j] +
      hb[f g, k, i, j] + CA[f g, j, k, l] - CA[f g, l, k, j]),
  pp[a[f_, i_, j_], δa[g_, k_, l_]] := Kδxi Kδyl
    (-δa[bk f g, i, j] + δa[bi f g, k, j]),
  pp[a[f_, i_, j_], δaa[g_, k_, l_, m_, n_]] := Kδxi (
    Kδyl (-δaa[bk f g, i, j, m, n] + δaa[bi f g, k, j, m, n]) +
    Kδyn (-δaa[bm f g, k, l, i, j] + δaa[bi f g, k, l, m, j]) +
    Kδyln (δa[bx bm f g, k, j] - δa[bk bm f g, x, j]),
  pp[_δβ, _a] → 0,
  pp[δa[f_, i_, j_], a[g_, k_, l_]] :=
    Kδxi Kδyl (-δa[bk f g, i, j] + δa[bi f g, k, j]),
  pp[δaa[f_, i_, j_, m_, n_], a[g_, k_, l_]] :=
    Kδxi Kδyl (-δaa[bk f g, i, j, m, n] + δaa[bi f g, k, j, m, n]) +
    Kδxm Kδyl (-δaa[bk f g, i, j, m, n] + δaa[bm f g, i, j, k, n])}]];
htb[x_, y_][L_UU, R_UU] := -thb[y, x][R, L];

```

$$t_1 h_1 t_2 h_2 \rightarrow t_1 t_2 h_1 h_2 \rightarrow t_2 t_1 h_1 h_2 \rightarrow t_2 t_1 h_2 h_1 \rightarrow t_2 h_2 t_1 h_1 :$$

db-def

```

db[x_][u_UU, v_UU] := Module[{t, h}, Plus[
  htb[x, x][u // tσ[x, t], v // hσ[x, h]] // tm[t, x, x] // hm[x, h, x],
  tb[x][u, v // hσ[x, h]] // hm[x, h, x],
  hb[x][u, v // tσ[x, t]] // tm[t, x, x],
  thb[x, x][u // hσ[x, h], v // tσ[x, t]] // tm[t, x, x] // hm[x, h, x] ]];

```

bb-def

```

bb[S_List] := Module[{w, bar, t, n = 0, i, k},
  w = #2 // dσ[S, bar/@S];
  Sum[t = db[S[[k]]][#1, w // dσ[bar[S[[k]], S[[k]]]];
  Do[t = t // dm[bar[S[[i]], S[[i]], S[[i]], {i, 1, k-1}];
  Do[t = t // dm[S[[i]], bar[S[[i]], S[[i]], {i, k+1, Length@S}];
  t, {k, Length@S} ] &
  bb[S_...] := bb[{S}]

```

ct (contract)

ct::usage =

"ct[h,t][L,R] contracts the head h in L with the tail t in R. ct[s][L,R] takes h=t=s, and ct[][L,R] takes s=0. When ambiguous, L is placed below R."

ct-def

```

ct[s_] := ct[s, s]; ct[] = ct[0, 0];
ct[h_, t_][UU[L_], UU[R_]] := CF[UU[Distribute[pp[L, R]] /. {
  pp[_β | _δβ, _] → 0,
  pp[a[f_, i_, h], β[g_]] := β[f bi ((∂btg) /. bt → 0)],
  pp[a[f_, i_, h], a[g_, t, j_]] := a[f (g /. bt → 0), i, j],
  pp[a[f_, i_, h], a[g_, j_, k_]] := a[f bi ((∂btg) /. bt → 0), j, k],
  pp[a[f_, i_, h], δa[g_, t, j_]] := δa[f (g /. bt → 0), i, j],
  pp[a[f_, i_, h], δa[g_, j_, k_]] := δa[f bi ((∂btg) /. bt → 0), j, k],
  pp[a[f_, i_, h], δaa[g_, t, j_, t, k_]] → 0,
  pp[a[f_, i_, h], δaa[g_, t, j_, k_, l_]] := δaa[f (g /. bt → 0), i, j, k, l],
  pp[a[f_, i_, h], δaa[g_, j_, k_, t, l_]] := δaa[f (g /. bt → 0), j, k, i, l],
  pp[a[f_, i_, h], δaa[g_, j_, k_, l_, m_]] :=
    δaa[f bi ((∂btg) /. bt → 0), j, k, l, m],
  pp[a[_], _] → 0, pp[_δa | _δaa, _δβ | _δa | _δaa] → 0,
  pp[δa[f_, i_, h], β[g_]] := δβ[f bi ((∂btg) /. bt → 0)],
  pp[δa[f_, i_, h], a[g_, t, j_]] := δa[f (g /. bt → 0), i, j],
  pp[δa[f_, i_, h], a[g_, j_, k_]] := δa[f bi ((∂btg) /. bt → 0), j, k],
  pp[_δa, _] → 0, pp[δaa[_], _, h, _, h], _] → 0,
  pp[δaa[f_, i_, h, j_, k_], β[g_]] := δa[f bi ((∂btg) /. bt → 0), j, k],
  pp[δaa[f_, i_, h, j_, k_], a[g_, t, l_]] := δaa[f (g /. bt → 0), i, l, j, k],
  pp[δaa[f_, i_, h, j_, k_], a[g_, l_, m_]] :=
    δaa[f bi ((∂btg) /. bt → 0), j, k, l, m],
  pp[δaa[f_, i_, j_, k_, h], β[g_]] := δa[f bk ((∂btg) /. bt → 0), i, j],
  pp[δaa[f_, i_, j_, k_, h], a[g_, t, l_]] := δaa[f (g /. bt → 0), i, j, k, l],
  pp[δaa[f_, i_, j_, k_, h], a[g_, l_, m_]] :=
    δaa[f bk ((∂btg) /. bt → 0), i, j, l, m],
  pp[_δaa, _] → 0 }]];

```

dect (de-contract)

```

dect::usage =
  "dect[h,t][uu] returns a pair {L,R} such that ct[h,t][L,R]=uu. Similarly
  for dect[s] and dect[]. uu is assumed to be atomic.";
dect[s_] := dect[s, s];
dect[] = dect[0, 0];
dect[h_, t_][β[f_]] := {};
dect[h_, t_][δβ[f_]] := TBD;

```

Exporting the above as PDF files

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

```

SetDirectory["C:\\drorbn\\AcademicPensieve\\Projects\\OneCo-1604"];

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
]

Button["Export",
  SetOptions[$FrontEndSession, PrintingStyleEnvironment -> "Working"];
  TagProperties[_] := {};
  TagProperties["ct-def"] = {PageWidth -> 6/0.65};
  Options[CellExport] = {
    PageWidth -> 4/0.65, 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 = Append[First[cells], PageWidth → 1.2 × 72 OptionValue[PageWidth]],  
  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}  
]  
]  
];  
nb = NotebookGet[EvaluationNotebook[]];  
tags = Cases[nb, (CellTags → tag_) ⇒ tag, Infinity] // Union;  
CellExport /@ tags;  
Print["Done."]  
]
```

Export

Exporting Snips\bb-def.pdf...
Exporting Snips\ct-def.pdf...
Exporting Snips\db-def.pdf...
Exporting Snips\delta-aa.pdf...
Exporting Snips\dm-def.pdf...
Exporting Snips\Generalities.pdf...
Exporting Snips\hb-def.pdf...
Exporting Snips\hm-def.pdf...
Exporting Snips\hts-def.pdf...
Exporting Snips\NonCommutativeMultiply.pdf...
Exporting Snips\sigma-def.pdf...
Exporting Snips\tb-def.pdf...
Exporting Snips\thb-def.pdf...
Exporting Snips\tm-def.pdf...
Done.