```
Print["MetaCalculi` loading..."]
```

Pensieve header: A common program for all w-meta-calculi. Continues pensieve://2014-07/MetaCalculi/.

General

```
Xp_{a_,b_} := Xp[a, b]; Xm_{a_,b_} := Xm[a, b];
```

```
SXForm[L_] := SXForm[
   Skeleton[L],
   Times @@ PD[L] /.
    X[i_, j_, k_, l_] :> If[PositiveQ[X[i, j, k, l]], Xp[l, i], Xm[j, i]]
  ];
Z[L_] := Z[Identity, L];
Z[χ_, L_] := Module[{s, z},
   {s, z} = List @@ SXForm[L];
   z = χ[z];
   Do[z = z // dm[s⟦c, 1⟧, s⟦c, k⟧ , s⟦c, 1⟧],
    {c, Length[s]}, {k, 2, Length[s⟦c⟧]}];
   z
  ];
```

```
dA[a_, rest__][α_] := α // dA[a] // dA[rest];
dA[l_List] := dA @@ l;
dA[All][α_] := α // dA[dL[α]];
dS[a_, rest__][α_] := α // dS[a] // dS[rest];
dS[l_List] := dS @@ l;
dS[All][α_] := α // dS[dL[α]];
```

α-Calculus

α-calculus is really the “exact” β-calculus of pensieve://2012-05/beta5.1

Utilities

```
αSimplify[expr_] := expr // Together // ExpandDenominator // ExpandNumerator;
SetAttributes[αCollect, Listable];
αCollect[A[ω_, σ_, μ_]] := A[
   αSimplify[ω], αSimplify[σ],
   Collect[μ, _h, Collect[#, _t, αSimplify] &]
  ];
αCollect[simp_][A[ω_, σ_, μ_]] := A[
   simp[ω], simp[σ],
   Collect[μ, _h, Collect[#, _t, simp] &]
  ];
hL[β_] := Union[Cases[β, h[s_] ⧴ s, ∞]];
tL[β_] := Union[Cases[β, t[s_] | c_s_ ⧴ s, ∞]];
dL[β_] := Union[hL[β], tL[β]];
A[ω_, σ_, μ_][Σ] := (∂_h[#] σ) & /@ hL[A[ω, σ, μ]];
αForm[A[ω_, σ_, μ_]] := Module[
   {tails, heads, M},
   tails = tL[A[ω, σ, μ]]; heads = hL[A[ω, σ, μ]];
   M = Outer[αSimplify[Coefficient[μ, h[#1] t[#2]]] &, heads, tails];
   PrependTo[M, t /@ tails];
   M = Prepend[Transpose[M], Prepend[h /@ heads, ω]];
   AppendTo[M, Prepend[A[ω, σ, μ][Σ], "A"]];
   MatrixForm[M]
  ];
αForm[else_] := else /. β_A ⧴ αForm[β];
Format[A[ω_, σ_, μ_], StandardForm] := αForm[A[ω, σ, μ]];
A /: A[ω1_, σ1_, μ1_] == A[ω2_, σ2_, μ2_] := Module[
  {heads, tails},
  tails = tL[{A[ω1, σ1, μ1], A[ω2, σ2, μ2]}];
  heads = hL[{A[ω1, σ1, μ1], A[ω2, σ2, μ2]}];
  (ω1 == ω2) ∧ (σ1 == σ2) ∧ (
    And @@ Flatten[Outer[
       (Coefficient[μ1, t[#1] h[#2]] == Coefficient[μ2, t[#1] h[#2]]) &,
       tails, heads
      ]]
   )
 ]
```

The Meta-Cross-Product

The “Tails” meta-group

```
tm[x_, y_, z_][β_A] := αCollect[β /. {t[x] → t[z], t[y] → t[z], c_x → c_z, c_y → c_z}];
tΔ[z_, x_, y_][β_A] := αCollect[β /. {t[z] → t[x] + t[y], c_z → c_x + c_y}];
tη[x_][β_A] := αCollect[(β /. t[x] → 0) /. c_x → 0];
tS[x_][β_A] := αCollect[β /. {t[x] → -t[x], c_x → -c_x}];
tA[_][β_A] := αCollect[β];
tσ[rules___Rule][β_A] := αCollect[
   β /. {t[x_] :→ t[x /. {rules}], c_x_ :→ c_(x /. {rules})}
  ];
```

The “Heads” meta-group

```
hm[x_, y_, z_][A[ω_, σ_, μ_]] := Module[
   {α = D[μ, h[x]], β = D[μ, h[y]], Ξ = μ /. h[x] | h[y] → 0},
   A[ω,
     h[z] (∂_h[x] σ) (∂_h[y] σ) + (σ /. h[x | y] → 0),
     Ξ + h[z] (α + (∂_h[x] σ) β)
    ] // αCollect
  ];
hΔ[z_, x_, y_][β_A] := αCollect[β /. h[z] → h[x] + h[y]];
hη[x_][β_A] := αCollect[β /. h[x] → 0];
hσ[rules___Rule][β_A] := αCollect[β /. h[x_] :→ h[x /. {rules}]];
```

The TH → HT and HT → TH Swaps

```
tha[u_, x_][A[ω_, σ_, μ_]] := Module[
   {α, θ, ϕ, Ξ, ν},
   α = Coefficient[μ, h[x] t[u]];
   θ = D[μ, t[u]] /. h[x] → 0;
   ϕ = D[μ, h[x]] /. t[u] → 0;
   Ξ = μ /. h[x] | t[u] → 0;
   ν = 1 + c_u α;
   A[ω * ν, σ, Plus[
      α (∂_h[x] σ) / ν * h[x] t[u],
      θ (∂_h[x] σ) / ν * t[u],
      ϕ / ν * h[x],
      Ξ - c_u ϕ * θ / ν
     ]] // αCollect
  ];
```

The “double” meta-group

```
dm[x_, y_, z_][β_A] := β // tha[x, y] // hm[x, y, z] // tm[x, y, z];
dΔ[z_, x_, y_][β_] := β // tΔ[z, x, y] // hΔ[z, x, y];
dη[s_][β_] := β // hη[s] // tη[s];
dσ[rules___][β_] := β // hσ[rules] // tσ[rules];
dσ[pl_List][β_] := Module[
   {σ, len, β1, k},
   len = Length[pl];
   β1 = β // (dσ @@ Table[i → σ[i], {i, len}]);
   Do[
    k = pl[[i, 1]];
    β1 = β1 // dσ[σ[i] → k];
    Do[
     β1 = β1 // dΔ[k, k, pl[[i, j]]],
     {j, 2, Length[pl[[i]]]}
    ],
    {i, len}
   ];
   β1
  ];
dσ[pl___Integer] := dσ[IntegerDigits /@ {pl}];
```

```
tr[a_][A[ω_, σ_, μ_]] := Module[
   {α, θ, ϕ, Ξ, ν},
   α = Coefficient[μ, h[a] t[a]];
   θ = D[μ, t[a]] /. h[a] → 0;
   ϕ = D[μ, h[a]] /. t[a] → 0;
   Ξ = μ /. h[a] | t[a] → 0;
   ν = (∂_h[a] σ) - 1 - c_a α;
   A[-ω * ν, σ /. h[a] → 0, Ξ + c_a ϕ θ / ν] // αCollect
  ];
```

The “external” product

```
A /: A[ω1_, σ1_, μ1_] A[ω2_, σ2_, μ2_] := A[ω1 * ω2, σ1 + σ2, μ1 + μ2];
```

Tangle Concatenation

```
A /: α1_A ** α2_A := Module[{S, α, τ},
  S = dL[{α1, α2}];
  α = α1 (α2 // dσ @@ ((# → τ[#]) & /@ S));
  α[[2]] += Total[h /@ Complement[S, hL[α1]]] + Total[h /@ τ /@ Complement[S, hL[α2]]];
  Do[
   α = α // dm[s, τ[s], s],
   {s, S}
  ];
  α
 ]
```

The R-Matrix

```
SetAttributes[A, Listable];
A[p_Times] := A /@ p;
A[Xp[a_, b_]] := αCollect[A[1, h[a] + e^(c_a) h[b], (e^(c_a) - 1) / c_a * t[a] h[b]]];
A[Xm[a_, b_]] := αCollect[A[1, h[a] + e^(-c_a) h[b], (e^(-c_a) - 1) / c_a * t[a] h[b]]];
```

Θ

```
A[Θ[a_, b_, p_]] := (A[1, e^(p c_a/2) h[1], (e^(p c_a/2) - 1) / c_a * t[a] h[a]] // dΔ[a, a, b]) **
   (A[1, e^(-p c_a/2) h[a], (e^(-p c_a/2) - 1) / c_a * t[a] h[a]]
     A[1, e^(-p c_b/2) h[b], (e^(-p c_b/2) - 1) / c_b * t[b] h[b]]);
A[Θ[a_, b_]] := A[Θ[a, b, 1]];
(*A[Θi[a_,b_]] := (A[1, (e^(-c_a/2)-1)/c_a*t[a]h[a]] // dΔ[a,a,b])**
  (A[1, (e^(c_a/2)-1)/c_a*t[a]h[a]]A[1, (e^(c_b/2)-1)/c_b*t[b]h[b]])*)
```

```
Γ[Θ[a_, b_]] := Θ[a, b] // A // Γ;
Γ[Θ[a_, b_, p_]] := Θ[a, b, p] // A // Γ;
```

The Exact KV Solution in α

See the 2014-07 version of this file.

Γ-Calculus

```
ΓSimp = Factor; SetAttributes[ΓCollect, Listable];
ΓCollect[Γ[ω_, σ_, λ_]] := ΓCollect[ΓSimp][Γ[ω, σ, λ]];
ΓCollect[simp_][Γ[ω_, σ_, λ_]] := Γ[simp[ω], simp[σ],
   Collect[λ, h_, Collect[#, t_, simp] &]];
dL[Γ[_, _, λ_]] := Union[Cases[λ, (h | t)a_ :> a, Infinity]];
Γ[ω1_, _, _][ω] := ω1;
Γ[ω_, σ_, λ_][Σ] := (∂h# σ) & /@ dL[Γ[ω, σ, λ]];
Γ[ω_, σ_, λ_][A] :=
  Module[{S = dL[Γ[ω, σ, λ]]}, Outer[ΓSimp[(∂t#1h#2 λ)] &, S, S]];
ΓForm[Γ[ω_, σ_, λ_]] := Module[{S, M},
   S = dL[Γ[ω, σ, λ]];
   M = Γ[ω, σ, λ][A] // Transpose;
   PrependTo[M, s# & /@ S];
   M = Join[
     {Prepend[s# & /@ S, ω]},
     Transpose[M],
     {Prepend[Γ[ω, σ, λ][Σ], "Γ"]}
    ];
   MatrixForm[M]
  ];
ΓForm[else_] := else /. Γ[ω_, σ_, λ_] :> ΓForm[Γ[ω, σ, λ]];
Format[Γ[ω_, σ_, λ_], StandardForm] := ΓForm[Γ[ω, σ, λ]];
```

```
Γ /: Γ[ω1_, σ1_, μ1_] == Γ[ω2_, σ2_, μ2_] := Module[
  {S},
  S = dL[Γ[ω1, σ1, μ1]] ⋃ dL[Γ[ω2, σ2, μ2]];
  (ω1 == ω2) && (And @@ ((∂h# σ1 == ∂h# σ2) & /@ S)) && (
    And @@ Flatten[Outer[
       (∂t#1,h#2 μ1 == ∂t#1,h#2 μ2) &,
       S, S
      ]]
   )
 ]
```

```
Γ /: Γ[ω1_, σ1_, λ1_] Γ[ω2_, σ2_, λ2_] := Γ[ω1 * ω2, σ1 + σ2, λ1 + λ2];
dm_{i_j_→k_}[Γ[ω_, σ_, λ_]] := Module[{α, β, γ, δ, θ, ϵ, ϕ, ψ, Ξ, μ},
   (α β θ; γ δ ϵ; ϕ ψ Ξ) = (∂_{t_i,h_i} λ  ∂_{t_i,h_j} λ  ∂_{t_i} λ;
                            ∂_{t_j,h_i} λ  ∂_{t_j,h_j} λ  ∂_{t_j} λ;
                            ∂_{h_i} λ      ∂_{h_j} λ      λ) /. (t | h)_{i|j} → 0;
   ΓCollect[Γ[(μ = 1 - β) ω,
       h_k (∂_{h_i} σ) (∂_{h_j} σ) + (σ /. h_{i|j} → 0),
       {t_k, 1}.(γ + α δ / μ   ϵ + δ θ / μ;
                 ϕ + α ψ / μ   Ξ + θ ψ / μ).{h_k, 1}
      ]] /. {T_i → T_k, T_j → T_k, b_i → b_k, b_j → b_k} // ΓCollect
  ];
dm[a_, b_, c_][Γ[ω_, σ_, λ_]] := dm_{ab→c}[Γ[ω, σ, λ]];
dη[a_][γ_Γ] := γ /. {(h | t)_a → 0, T_a → 1};
```

```
tr[a_][Γ[ω_, σ_, λ_]] := Module[{α, θ, ψ, Ξ},
   (α θ; ψ Ξ) = (∂_{t_a,h_a} λ  ∂_{t_a} λ;
                 ∂_{h_a} λ      λ) /. (t | h)_a → 0;
   Γ[ω (1 - α), σ /. h_a → 0, Ξ + ψ * θ / (1 - α)] // ΓCollect];
```

```
FullStitch[γ1_Γ, γ2_Γ] := Module[{S1, S2, S, γ, τ},
   S = (S1 = dL[γ1]) ⋃ (S2 = dL[γ2]);
   γ = γ1 (Times @@ (Γ /@ ϵ /@ Complement[S, S1]));
   γ *= (γ2 /. {h_a_ :→ h_{τ[a]}, t_a_ :→ t_{τ[a]}, T_a_ :→ T_{τ[a]}})
      (Times @@ (Γ /@ ϵ /@ τ /@ Complement[S, S2]));
   Do[
    γ = γ // dm[s, τ[s], s],
    {s, S}
   ];
   γ
  ];
Γ /: γ1_Γ ** γ2_Γ := Module[{S1, S2, S, γ1p, γ2p},
   S = (S1 = dL[γ1]) ⋃ (S2 = dL[γ2]);
   γ1p = γ1 (Times @@ (Γ /@ ϵ /@ Complement[S, S1]));
   γ2p = γ2 (Times @@ (Γ /@ ϵ /@ Complement[S, S2]));
   Γ[
    γ1p[ω] * γ2p[ω],
    (γ1p[Σ] γ2p[Σ]).(h_# & /@ S),
    (t_# & /@ S).(γ2p[A].γ1p[A]).(h_# & /@ S)
   ]
  ];
```

```
Γ /: Γ[ω_, σ_, λ_]^-1 := Module[{S = dL[Γ[ω, σ, λ]]},
   Γ[
    ω^-1, Collect[σ, h_, (1/#) &],
    (t_# & /@ S).Inverse[Outer[ΓSimp[(∂_{t_#1 h_#2} λ)] &, S, S]].(h_# & /@ S)
   ]
  ];
```

```
dA[a_][Γ[ω_, σ_, λ_]] := Module[
   {α, θ, ϕ, Ξ, σa},
   ( α  θ ) = ( ∂_{t_a,h_a} λ  ∂_{t_a} λ ) /. (h | t)_a → 0;
   ( ϕ  Ξ )   ( ∂_{h_a} λ      λ        )
   σa = ∂_{h_a} σ;
   ΓCollect[Γ[
     α ω / σa,
     ((σ /. h_a → 0) + h_a / σa),
     {t_a, 1}.( 1    θ         ).{h_a, 1} / α
              ( -ϕ   α Ξ - ϕ θ )
    ]]
  ];
dS[a_][γ_Γ] := ΓCollect[dA[a][γ] /. {T_a → 1/T_a, b_a → -b_a}];
```

```
Mirror[γ_Γ] := Module[{γ1},
   γ1 = γ // (dS @@ dL[γ]);
   γ1[[3]] = γ1[[3]] /. {t_a_ :→ h_a, h_a_ :→ t_a};
   γ1];
```

```
tσ[rules___Rule][γ_Γ] := ΓCollect[
   γ /. {t_u_ :→ t_{u /. {rules}}, T_u_ :→ T_{u /. {rules}}, b_u_ :→ b_{u /. {rules}}}
  ];
hσ[rules___Rule][γ_Γ] := ΓCollect[γ /. h_x_ :→ h_{x /. {rules}}];
```

```
SetAttributes[Γ, Listable];
Γ[p_Times | p_NonCommutativeMultiply] := Γ /@ p;
Γ[ϵ[a_]] := Γ[1, h_a, h_a t_a];
Γ[Xp[a_, b_]] := Γ[1, h_a + h_b T_a, {t_a, t_b}.( 1  1 - T_a ).{h_a, h_b}];
                                                ( 0  T_a     )
Γ[Xm[a_, b_]] := Γ[Xp[a, b]] /. T_a → 1/T_a;
```

```
MVA[Γ, L_Link] := Module[{Hs, ω, σ, μ, A},
  {ω, σ, μ} = List @@ Z[Γ, L];
  Hs = Rest[h# & /@ (First /@ Skeleton[L])];
  A = Outer[Coefficient[μ, #1 * #2] &, Hs, Hs /. h_a_ :> t_a];
  Factor[(ω Det[A - IdentityMatrix@Length@Hs]) / (1 - T_Skeleton[L][[1,1]])]
 ]
```

dΔ (“naive cabling”)

```
dΔ[a_, x_, y_][Γ[ω_, σ_, λ_]] := Module[
   {α, θ, ϕ, Ξ, σa, Ta, M},
   {{α, θ}, {ϕ, Ξ}} = {{∂_{t_a,h_a} λ, ∂_{t_a} λ}, {∂_{h_a} λ, λ}} /. (h | t)_a → 0 /. T_a → Ta;
   σa = ∂_{h_a} σ /. T_a → Ta;
   M = {{α + Log[T_y] (-α+σa)/Log[Ta], Log[T_x] (α-σa)/Log[Ta], θ Log[T_x]/Log[Ta]},
        {Log[T_y] (α-σa)/Log[Ta], α + Log[T_x] (-α+σa)/Log[Ta], θ Log[T_y]/Log[Ta]},
        {ϕ, ϕ, Ξ}};
   ΓCollect[Γ[
     ω /. T_a → T_x T_y,
     ((σ /. h_a → 0) + (h_x + h_y) σa) /. Ta | T_a → T_x T_y,
     {t_x, t_y, 1}.M.{h_x, h_y, 1} /. Ta → T_x T_y
    ]]
  ];
```

qΔ (“renormalized cabling”)

```
qΔ[a_, x_, y_][Γ[ω_, σ_, λ_]] := Module[
  {α, θ, ϕ, Ξ, σa, Ta, M},
  (α θ; ϕ Ξ) = (∂_{t_a,h_a} λ  ∂_{t_a} λ; ∂_{h_a} λ  λ) /. (h | t)_a → 0 /. T_a → Ta;
  σa = ∂_{h_a} σ /. T_a → Ta;
  M = ( (-σa+α Ta+(-α+σa) T_y)/(-1+Ta)   ((-1+T_x) T_y (α-σa))/(-1+Ta)   (θ (-1+T_x) T_y)/(-1+Ta);
        ((-1+T_y) (α-σa))/(-1+Ta)        (-α+σa Ta+(α-σa) T_y)/(-1+Ta)   (θ (-1+T_y))/(-1+Ta);
        ϕ                                 ϕ                               Ξ );
  ΓCollect[Γ[
    ω /. T_a → T_x T_y,
    ((σ /. h_a → 0) + (h_x + h_y) σa) /. T_a | Ta → T_x T_y,
    {t_x, t_y, 1}.M.{h_x, h_y, 1} /. Ta → T_x T_y
   ]]
 ];
```

The Exact KV Solution in Γ

Probably coming from pensieve://2012-05/beta5.1/betaExact.nb.

$$\Gamma[V] = \Gamma\Bigg[\frac{\left(\frac{-1+T_1}{\mathrm{Log}[T_1]}\right)^{1/4}\left(\frac{-1+T_2}{\mathrm{Log}[T_2]}\right)^{1/4}}{\left(\frac{-1+T_1 T_2}{\mathrm{Log}[T_1 T_2]}\right)^{1/4}},\ h_1 + h_2\sqrt{T_1},$$

$$\{t_1, t_2\}.\begin{pmatrix} \frac{\mathrm{Log}[T_1]\left(1+\frac{\mathrm{Log}[T_2]\sqrt{\frac{(-1+T_1)T_1}{\mathrm{Log}[T_1]}}\sqrt{\frac{-1+T_2}{\mathrm{Log}[T_2]}}}{(-1+T_1)\sqrt{\frac{-1+T_1 T_2}{\mathrm{Log}[T_1 T_2]}}}\right)}{\mathrm{Log}[T_1 T_2]} & \frac{\mathrm{Log}[T_1]\left(1-\frac{\sqrt{\frac{(-1+T_1)T_1}{\mathrm{Log}[T_1]}}\,T_2}{\sqrt{\frac{-1+T_2}{\mathrm{Log}[T_2]}}\sqrt{\frac{-1+T_1 T_2}{\mathrm{Log}[T_1 T_2]}}}\right)}{\mathrm{Log}[T_1 T_2]} \\ \frac{\mathrm{Log}[T_2]\left(1-\frac{T_1\sqrt{\frac{-1+T_2}{\mathrm{Log}[T_2]}}}{\sqrt{\frac{(-1+T_1)T_1}{\mathrm{Log}[T_1]}}\sqrt{\frac{-1+T_1 T_2}{\mathrm{Log}[T_1 T_2]}}}\right)}{\mathrm{Log}[T_1 T_2]} & \frac{\mathrm{Log}[T_2]\left(1+\frac{\mathrm{Log}[T_1]\sqrt{\frac{(-1+T_1)T_1}{\mathrm{Log}[T_1]}}\,T_2}{\mathrm{Log}[T_2]\sqrt{\frac{-1+T_2}{\mathrm{Log}[T_2]}}\sqrt{\frac{-1+T_1 T_2}{\mathrm{Log}[T_1 T_2]}}}\right)}{\mathrm{Log}[T_1 T_2]} \end{pmatrix}.\{h_1, h_2\}$$

$$\Bigg];$$

$$\Gamma[Vi] = \Gamma\Bigg[\frac{\left(\frac{-1+T_1 T_2}{\mathrm{Log}[T_1 T_2]}\right)^{1/4}}{\left(\frac{-1+T_1}{\mathrm{Log}[T_1]}\right)^{1/4}\left(\frac{-1+T_2}{\mathrm{Log}[T_2]}\right)^{1/4}},\ h_1 + h_2\Big/\sqrt{T_1},$$

$$\{t_1, t_2\}.\begin{pmatrix} \frac{(-1+T_1)T_2+\mathrm{Log}[T_2]\sqrt{\frac{-1+T_1}{\mathrm{Log}[T_1]\,T_1}}\sqrt{\frac{-1+T_2}{\mathrm{Log}[T_2]}}\sqrt{\frac{-1+T_1 T_2}{\mathrm{Log}[T_1 T_2]}}}{-1+T_1 T_2} & \frac{(-1+T_1)T_2-\mathrm{Log}[T_1]\sqrt{\frac{-1+T_1}{\mathrm{Log}[T_1]\,T_1}}\sqrt{\frac{-1+T_2}{\mathrm{Log}[T_2]}}\sqrt{\frac{-1+T_1 T_2}{\mathrm{Log}[T_1 T_2]}}}{-1+T_1 T_2} \\ \frac{-1+T_2-\mathrm{Log}[T_2]\sqrt{\frac{-1+T_1}{\mathrm{Log}[T_1]\,T_1}}\sqrt{\frac{-1+T_2}{\mathrm{Log}[T_2]}}\sqrt{\frac{-1+T_1 T_2}{\mathrm{Log}[T_1 T_2]}}}{-1+T_1 T_2} & \frac{-1+T_2+\mathrm{Log}[T_1]\sqrt{\frac{-1+T_1}{\mathrm{Log}[T_1]\,T_1}}\sqrt{\frac{-1+T_2}{\mathrm{Log}[T_2]}}\sqrt{\frac{-1+T_1 T_2}{\mathrm{Log}[T_1 T_2]}}}{-1+T_1 T_2} \end{pmatrix}.$$

$$\{h_1, h_2\}$$

$$\Bigg];$$

α↔Γ Conversions

```
A[ω_, σ_, μ_] // Γ := Module[{S},
   S = dL[{ω, μ}];
   Γ[ω,
      Total[h# & /@ S] + μ /. {h[a_] ⧴ h_a, t[a_] ⧴ c_a},
      Total[(t# h#) & /@ S] + (μ /. t[a_] ⧴ c_a /. h[a_] ⧴ h_a t_a) - μ /.
       {h[a_] ⧴ h_a, t[a_] ⧴ c_a t_a}
     ] /. c_a_ ⧴ Log[T_a] // ΓCollect[FullSimplify[# /. Sinh[x_] ⧴ (e^x - e^-x)/2] &]
  ];
```

```
Γ[ω_, σ_, λ_] // A := Module[{S, μ},
   S = dL[Γ[ω, σ, λ]];
   μ = Total[((∂h# σ) t# h#) & /@ S] - λ;
   A[ω,
      σ /. ha_ :> h[a],
      μ /. {ha_ :> h[a], ta_ :> t[a] / ca}
     ] /. Ta_ :> ⅇ^ca // αCollect
  ];
```

Γb-Calculus

```
ΓbSimp = Factor; SetAttributes[ΓbCollect, Listable];
ΓbCollect[Γb[ω_, σ_, λ_]] := ΓbCollect[ΓbSimp][Γb[ω, σ, λ]];
ΓbCollect[simp_][Γb[ω_, σ_, λ_]] := Γb[simp[ω], simp[σ],
   Collect[λ, h_, Collect[#, t_, simp] &]];
dL[Γb[_, _, λ_]] := dL[Γ[1, 1, λ]];
Γb[ω1_, _, _][ω] := ω1;
Γb[ω_, σ_, λ_][Σ] := (∂h# σ) & /@ dL[Γb[ω, σ, λ]];
Γb[ω_, σ_, λ_][A] :=
  Module[{S = dL[Γb[ω, σ, λ]]}, Outer[ΓbSimp[(∂t#1h#2 λ)] &, S, S]];
ΓbForm[Γb[ω_, σ_, λ_]] := Module[{S, M},
   S = dL[Γb[ω, σ, λ]];
   M = Γb[ω, σ, λ][A] // Transpose;
   PrependTo[M, s# & /@ S];
   M = Join[
     {Prepend[s# & /@ S, ω]},
     Transpose[M],
     {Prepend[Γb[ω, σ, λ][Σ], "Σ"]}
    ];
   MatrixForm[M]
  ];
ΓbForm[else_] := else /. Γb[ω_, σ_, λ_] :> ΓbForm[Γb[ω, σ, λ]];
Format[Γb[ω_, σ_, λ_], StandardForm] := ΓbForm[Γb[ω, σ, λ]];
```

```
Γ[ω_, σ_, λ_] // Γb := ΓbCollect[Γb[ω, σ, ω * λ]];
Γb[ω_, σ_, λ_] // Γ := ΓCollect[Γ[ω, σ, λ / ω]];
α_A // Γb := α // Γ // Γb
```

Γ1-Calculus

```
Γ1Simp = Factor; SetAttributes[Γ1Collect, Listable];
Γ1Collect[Γ1[ω_, σ_, λ_]] := Γ1Collect[Γ1Simp][Γ1[ω, σ, λ]];
Γ1Collect[simp_][Γ1[ω_, σ_, λ_]] := Γ1[simp[ω], simp[σ],
   Collect[λ, h_, Collect[#, t_, simp] &]];
dL[Γ1[_, _, λ_]] := dL[Γ[1, 1, λ]];
Γ1[ω1_, _, _][ω] := ω1;
Γ1[ω_, σ_, λ_][Σ] := (∂_{h_#} σ) & /@ dL[Γ1[ω, σ, λ]];
Γ1[ω_, σ_, λ_][A] :=
  Module[{S = dL[Γb[ω, σ, λ]]}, Outer[ΓbSimp[(∂_{t_#1 h_#2} λ)] &, S, S]];
Γ1Form[Γ1[ω_, σ_, λ_]] := Module[{S, M},
   S = dL[Γ1[ω, σ, λ]];
   M = Γ1[ω, σ, λ][A] // Transpose;
   PrependTo[M, s_# & /@ S];
   M = Join[
     {Prepend[s_# & /@ S, ω]},
     Transpose[M],
     {Prepend[Γ1[ω, σ, λ][Σ], "Γ1"]}
    ];
   MatrixForm[M]
  ];
Γ1Form[else_] := else /. Γ1[ω_, σ_, λ_] :> Γ1Form[Γ1[ω, σ, λ]];
Format[Γ1[ω_, σ_, λ_], StandardForm] := Γ1Form[Γ1[ω, σ, λ]];
```

```
Γ[ω_, σ_, λ_] // Γ1 :=
  Γ1Collect[Γ1[ω, σ, λ /. {h_s_ :> (1 - T_s) h_s, t_s_ :> t_s / (1 - T_s)}]];
Γ1[ω_, σ_, λ_] // Γ := ΓCollect[Γ[ω, σ, λ /. {h_s_ :> h_s / (1 - T_s), t_s_ :> (1 - T_s) t_s}]];
α_A // Γ1 := α // Γ // Γ1
```

```
SetAttributes[Γ1, Listable];
Γ1[p_Times | p_NonCommutativeMultiply] := Γ1 /@ p;
Γ1[ϵ[a_]] := Γ1[1, h_a, h_a t_a];
Γ1[Xp[a_, b_]] := Xp[a, b] // Γ // Γ1;
Γ1[Xm[a_, b_]] := Xm[a, b] // Γ // Γ1;
```

```
Γ1 /: Γ1[ω1_, σ1_, λ1_] Γ1[ω2_, σ2_, λ2_] := Γ1[ω1 * ω2, σ1 + σ2, λ1 + λ2];
γ_Γ1 // op_dm := γ // Γ // op // Γ1;
γ_Γ1 // op_qΔ := γ // Γ // op // Γ1;
```