

Pensieve header: A common program for all w-meta-calculi. Continued pensieve://2014-06/MetaCalculi/.

---

## General

```

Xpa,b := Xp[a, b]; Xma,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
];

```

---

## $\alpha$ -Calculus

$\alpha$ -calculus is really the “exact”  $\beta$ -calculus of pensieve://2012-05/beta5.1

## Utilities

```

αSimplify[expr_] := expr // Together // ExpandDenominator // ExpandNumerator;
SetAttributes[αCollect, Listable];
αCollect[A[ω_, μ_]] := A[
  αSimplify[ω],
  Collect[μ, _h, Collect[#, _t, αSimplify] &]
];
αCollect[simp_][A[ω_, μ_]] := A[
  simp[ω],
  Collect[μ, _h, Collect[#, _t, simp] &]
];
hL[β_] := Union[Cases[β, h[s_] => s, Infinity]];
tL[β_] := Union[Cases[β, t[s_] | c_s_ => s, Infinity]];
dL[β_] := Union[hL[β], tL[β]];
αForm[A[ω_, μ_]] := Module[
  {tails, heads, mat},
  tails = tL[A[ω, μ]]; heads = hL[A[ω, μ]];
  mat = Outer[αSimplify[Coefficient[μ, h[#1] t[#2]]] &, heads, tails];
  PrependTo[mat, t /@ tails];
  mat = Prepend[Transpose[mat], Prepend[h /@ heads, ω]];
  MatrixForm[mat]
];
αForm[else_] := else /. β_A => αForm[β];
Format[A[ω_, μ_], StandardForm] := αForm[A[ω, μ]];
A /: A[ω1_, μ1_] == A[ω2_, μ2_] := Module[
  {heads, tails},
  tails = tL[{A[ω1, μ1], A[ω2, μ2]}];
  heads = hL[{A[ω1, μ1], A[ω2, μ2]}];
  (ω1 == ω2) && (
    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_, μ_]] := Module[
  {γx = D[μ, h[x]], γy = D[μ, h[y]], M = μ /. h[x] | h[y] → 0,
  A[ω, M + h[z] (γx + γy + (γx /. t[i_] ⇒ c_i) γy)] // αCollect
];
hΔ[z_, x_, y_][β_A] := αCollect[β /. h[z] → h[x] + h[y]];
hη[x_][β_A] := αCollect[β /. h[x] → 0];
hS[x_][A[ω_, μ_]] := Module[{γ},
  γ = 1 + D[μ, h[x]] /. t[s_] ⇒ c_s;
  αCollect[A[ω, μ /. h[x] → -h[x] / γ]]
];
hA[x_][β_A] := hS[x][β];
hσ[rules__Rule][β_A] := αCollect[β /. h[x_] ⇒ h[x /. {rules}]];
```

The TH → HT and HT → TH Swaps

```
tha[x_, y_][A[ω_, μ_]] := Module[
  {α, β, γ, δ, ε},
  α = Coefficient[μ, h[y] t[x]];
  β = D[μ, t[x]] /. h[y] → 0;
  γ = D[μ, h[y]] /. t[x] → 0;
  δ = μ /. h[y] | t[x] → 0;
  ε = 1 + c_x α;
  A[ω * ε, Plus[
    α (1 + (γ /. t[i_] ⇒ c_i) / ε) h[y] t[x],
    β (1 + (γ /. t[i_] ⇒ c_i) / ε) t[x],
    γ / ε h[y],
    δ - c_x / ε γ * β
  ]] // αCollect
];
hta[x_, y_][β_A] := β // hS[x] // tha[y, x] // hS[x];
```

## The “double” meta-group

```

dm[x_, y_, z_][β_] := 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];
ds[s_][β_] := β // hta[s, s] // hs[s] // ts[s];
dA[s_][β_] := β // hta[s, s] // hA[s] // tA[s];
dη[s_][β_] := β // hη[s] // tη[s];
dcap[s_][β_] := β // hta[s, s] // hη[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}];

```

## The “external” product

```

A /: A[ω1_, μ1_] A[ω2_, μ2_] := A[ω1 * ω2, μ1 + μ2];

```

## Tangle Concatenation

```

A /: α1_A ** α2_A := Module[{S, α, τ},
  S = dL[α1];
  α = α1 (α2 // dσ@@ (# → τ[#]) & /@ S);
  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, (e^ca - 1) / ca * t[a] h[b]]];
A[Xm[a_, b_]] := αCollect[A[1, (e^-ca - 1) / ca * t[a] h[b]]];
A[Θ[a_, b_]] := (A[1, (e^ca/2 - 1) / ca * t[a] h[a]] // dΔ[a, a, b]) **
    (A[1, (e^-ca/2 - 1) / ca * t[a] h[a]] A[1, (e^-cb/2 - 1) / cb * t[b] h[b]]);
A[Θi[a_, b_]] := (A[1, (e^-ca/2 - 1) / ca * t[a] h[a]] // dΔ[a, a, b]) **
    (A[1, (e^ca/2 - 1) / ca * t[a] h[a]] A[1, (e^cb/2 - 1) / cb * t[b] h[b]])
    
```

### The Exact KV Solution in α

```

Module[{v, κ, ω, α, β, γ, δ},
    
```

$$v[x_] := \sqrt{\frac{\sinh\left[\frac{x}{2}\right]}{x/2}}; \quad \kappa[x_] := v[x]^{-1/2}; \quad \omega = \frac{\kappa[c_1 + c_2]}{\kappa[c_1] \kappa[c_2]};$$

$$\gamma = \frac{v[c_2] - v[c_1] v[c_1 + c_2]}{(c_1 + c_2) v[c_1] v[c_1 + c_2]}; \quad \delta = \frac{e^{\frac{c_1}{2}}}{c_2} - \frac{v[c_1 + c_2] e^{c_1 + c_2} v[c_1] c_1}{(-1 + e^{c_1 + c_2}) v[c_2] c_2} - \frac{1}{c_1 + c_2};$$

$$\alpha = \frac{-c_2}{c_1} \gamma; \quad \beta = \frac{1}{c_1} \left( e^{\frac{c_1}{2}} - c_2 \delta - 1 \right);$$

```

{A[C] = αCollect[A[κ[c_1], 0]],
    
```

```

A[V] = αCollect[A[ω, {t@1, t@2}. {α β}. {γ δ}. {h@1, h@2}]],
    
```

```

A[Vi] = A[V] // dA[1] // dA[2]
}
]
    
```

$$\left\{ \begin{array}{l} \frac{1}{2^{1/4} \left( \frac{\sinh\left[\frac{c_1}{2}\right]}{c_1} \right)^{1/4}} \\ t[1] \\ t[2] \end{array} \right\}, \quad \left( \begin{array}{l} \frac{2^{1/4} \left( \frac{\sinh\left[\frac{c_1}{2}\right]}{c_1} \right)^{1/4} \left( \frac{\sinh\left[\frac{c_2}{2}\right]}{c_2} \right)^{1/4}}{\left( \frac{\sinh\left[\frac{1}{2}(c_1+c_2)}\right]}{c_1+c_2} \right)^{1/4}} \\ h[1] \\ t[1] \\ t[2] \end{array} \right)$$

$$\frac{-\sqrt{2} \sqrt{\frac{\sinh\left[\frac{c_2}{2}\right]}{c_2}} c_2 + 2 \sqrt{\frac{\sinh\left[\frac{c_1}{2}\right]}{c_1}} c_2 \sqrt{\frac{\sinh\left[\frac{1}{2}(c_1+c_2)}\right]}{c_1+c_2}}}{2 \sqrt{\frac{\sinh\left[\frac{c_1}{2}\right]}{c_1}} c_1^2 \sqrt{\frac{\sinh\left[\frac{1}{2}(c_1+c_2)}\right]}{c_1+c_2}} + 2 \sqrt{\frac{\sinh\left[\frac{c_1}{2}\right]}{c_1}} c_1 c_2 \sqrt{\frac{\sinh\left[\frac{1}{2}(c_1+c_2)}\right]}{c_1+c_2}}}$$

$$\frac{\sqrt{2} \sqrt{\frac{\sinh\left[\frac{c_2}{2}\right]}{c_2}} - 2 \sqrt{\frac{\sinh\left[\frac{c_1}{2}\right]}{c_1}} \sqrt{\frac{\sinh\left[\frac{1}{2}(c_1+c_2)}\right]}{c_1+c_2}}}{2 \sqrt{\frac{\sinh\left[\frac{c_1}{2}\right]}{c_1}} c_1 \sqrt{\frac{\sinh\left[\frac{1}{2}(c_1+c_2)}\right]}{c_1+c_2}} + 2 \sqrt{\frac{\sinh\left[\frac{c_1}{2}\right]}{c_1}} c_2 \sqrt{\frac{\sinh\left[\frac{1}{2}(c_1+c_2)}\right]}{c_1+c_2}}} \quad -e^{\frac{c_1}{2}} c$$

$q\Delta$  (“renormalized cabling”)

```

qDelta[a_, x_, y_][A[w_, mu_]] := Module[
  {alpha, theta, phi, xi, M},
  (
    alpha  theta
    phi  xi
  ) = (
    partial_h[a], t[a] mu  partial_t[a] mu
    partial_h[a] mu      mu
  ) /. (h | t)[a] -> 0;

  M = (
    ( (e^{c_x} - e^{c_y}) alpha c_a / (-1 + e^{c_x}) c_x, (e^{c_x} - e^{c_y}) alpha c_a / (-1 + e^{c_x}) c_x, (e^{c_x} - e^{c_y}) theta c_a / (-1 + e^{c_x}) c_x,
    ( (-1 + e^{c_y}) alpha c_a / (-1 + e^{c_x}) c_y, (-1 + e^{c_y}) alpha c_a / (-1 + e^{c_x}) c_y, (-1 + e^{c_y}) theta c_a / (-1 + e^{c_x}) c_y,
    phi, phi, xi
  );

  alphaCollect[A[w, {t[x], t[y], 1}.M.{h[x], h[y], 1} /. c_a -> c_x + c_y]
];

```

## $\Gamma$ -Calculus

```

GammaSimp = Factor; SetAttributes[GammaCollect, Listable];
GammaCollect[Gamma[w_, sigma_, lambda_]] := GammaCollect[GammaSimp][Gamma[w, sigma, lambda]];
GammaCollect[simp_][Gamma[w_, sigma_, lambda_]] := Gamma[simp[w], simp[sigma],
  Collect[lambda, h_, Collect[#, t_, simp] &]];
dL[Gamma[_ , _ , lambda_]] := Union[Cases[lambda, (h | t)_a -> a, Infinity]];
Gamma[w1_, _ , _][w] := w1;
Gamma[w_, sigma_, lambda_][Sigma] := (partial_h[sigma] & /@ dL[Gamma[w, sigma, lambda]]);
Gamma[w_, sigma_, lambda_][A] :=
  Module[{S = dL[Gamma[w, sigma, lambda]], Outer[GammaSimp[(partial_t[h1] h2 lambda)] &, S, S]};
GammaForm[Gamma[w_, sigma_, lambda_]] := Module[{S, M},
  S = dL[Gamma[w, sigma, lambda]];
  M = Gamma[w, sigma, lambda][A] // Transpose;
  PrependTo[M, s# & /@ S];
  M = Join[
    {Prepend[s# & /@ S, w]},
    Transpose[M],
    {Prepend[Gamma[w, sigma, lambda][Sigma], "Sigma"]}
  ];
  MatrixForm[M]
];
GammaForm[else_] := else /. Gamma[w_, sigma_, lambda_] -> GammaForm[Gamma[w, sigma, lambda]];
Format[Gamma[w_, sigma_, lambda_], StandardForm] := GammaForm[Gamma[w, sigma, lambda]];

```

```

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 @@ ((∂ha σ1 == ∂ha σ2) & /@ S)) && (
    And @@ Flatten[Outer[
      (∂ta1, ha2 μ1 == ∂ta1, ha2 μ2) &,
      S, S
    ]]
  )
]

Γ /: Γ[ω1_, σ1_, λ1_] Γ[ω2_, σ2_, λ2_] := Γ[ω1*ω2, σ1+σ2, λ1+λ2];
dma_b→c[Γ[ω_, σ_, λ_]] := Module[{α, β, γ, δ, θ, ε, φ, ψ, Ξ, μ},
  
$$\begin{pmatrix} \alpha & \beta & \theta \\ \gamma & \delta & \epsilon \\ \phi & \psi & \Xi \end{pmatrix} = \begin{pmatrix} \partial_{t_a, h_a} \lambda & \partial_{t_a, h_b} \lambda & \partial_{t_a} \lambda \\ \partial_{t_b, h_a} \lambda & \partial_{t_b, h_b} \lambda & \partial_{t_b} \lambda \\ \partial_{h_a} \lambda & \partial_{h_b} \lambda & \lambda \end{pmatrix} /. (t | h)_{a|b} \rightarrow 0;$$

  ΓCollect[Γ[(μ = 1 - β) ω,
    hc (∂ha σ) (∂hb σ) + (σ /. ha|b → 0),
    {tc, 1} . {γ + α δ / μ, ε + δ θ / μ} . {hc, 1}
  ]] /. {Ta → Tc, Tb → Tc} // ΓCollect
];
dm[a_, b_, c_][Γ[ω_, σ_, λ_]] := dmab→c[Γ[ω, σ, λ]];
dη[a_][γ_Γ] := γ /. {(h | t)a → 0, Ta → 1};
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 /. {ha → hτ[a], ta → tτ[a], Ta → Tτ[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, σα},
  (α θ) = (∂_{t_a, h_a}λ ∂_{t_a}λ) /. (h | t)_a → 0;
  (φ Ξ) = (∂_{h_a}λ λ) /. (h | t)_a → 0;
  σα = ∂_{h_a}σ;
  ΓCollect[Γ[
    α ω / σα,
    ((σ /. h_a → 0) + h_a / σα),
    {t_a, 1}.(1 θ) / (-φ α Ξ - φ θ) . {h_a, 1} / α
  ]
];

ds[a_][γ_Γ] := ΓCollect[dA[a][γ] /. T_a → 1 / T_a];

to[rules__Rule][γ_Γ] := ΓCollect[
  γ /. {t_u_ :=> t_u /. {rules}, T_u_ :=> T_u /. {rules}}
];

ho[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) / (0 T_a) . {h_a, h_b}];
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]])]
];

```



qΔ (“renormalized cabling”)

```

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

```

## The Exact KV Solution in $\Gamma$

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

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

];

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

$\{t_1, t_2\} \cdot$

$$\left( \begin{array}{cc} \frac{(-1+t_1) T_2 + \text{Log}[T_2] \sqrt{\frac{-1+t_1}{\text{Log}[t_1] T_1}} \sqrt{\frac{-1+t_2}{\text{Log}[t_2]}} \sqrt{\frac{-1+t_1 T_2}{\text{Log}[t_1 T_2]}}}{-1+t_1 T_2} & \frac{(-1+t_1) T_2 - \text{Log}[T_1] \sqrt{\frac{-1+t_1}{\text{Log}[t_1] T_1}} \sqrt{\frac{-1+t_2}{\text{Log}[t_2]}} \sqrt{\frac{-1+t_1 T_2}{\text{Log}[t_1 T_2]}}}{-1+t_1 T_2} \\ \frac{-1+T_2 - \text{Log}[T_2] \sqrt{\frac{-1+t_1}{\text{Log}[t_1] T_1}} \sqrt{\frac{-1+t_2}{\text{Log}[t_2]}} \sqrt{\frac{-1+t_1 T_2}{\text{Log}[t_1 T_2]}}}{-1+t_1 T_2} & \frac{-1+T_2 + \text{Log}[T_1] \sqrt{\frac{-1+t_1}{\text{Log}[t_1] T_1}} \sqrt{\frac{-1+t_2}{\text{Log}[t_2]}} \sqrt{\frac{-1+t_1 T_2}{\text{Log}[t_1 T_2]}}}{-1+t_1 T_2} \end{array} \right) \cdot$$

$\{h_1, h_2\}$

];

## $\alpha \leftrightarrow \Gamma$ Conversions

```
A[ω_, 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] // ICollect
];
```

```

Γ[ω_, σ_, λ_] // A := Module[{S, μ},
  S = dL[Γ[ω, σ, λ]];
  μ = Total[((∂h# σ) t# h#) & /@ S] - λ;
  A[ω,
    μ /. {ha -> h[a], ta -> t[a] / ca}
  ] /. Ta -> eca // αCollect
];

```