Pensieve header: The Engine, with Zip3 encapsulation.

pdf

The Engine

pdf

Canonical Forms:

pdf

```
CCF[ℰ_] := PP_CCF@Factor[ℰ]; (*Coefficient Canonical Form *)
LogReduce[ℰ_] := ℰ /. c_ * Log[a_] :> Log@Factor[a^c] //. Log[a_] + Log[b_] :> Log@Factor[a b];
CF[ℰ_] := PP_CF@Module[
   {vs = Cases[ℰ, (y | a | x | η | β | τ | ξ)_, ∞] ⋃ {y, a, x, η, β, τ, ξ}},
   Total[(CCF[#[[2]]] (Times @@ vs^#[[1]])) & /@ CoefficientRules[ℰ, vs]]
  ];
CF[ℰ_𝔼] := CF /@ MapAt[LogReduce, ℰ, 1];
CF[U_𝕌] := CF /@ MapAt[LogReduce, U, 1];
CF[ℰ_List] := CF /@ ℰ;
CF[𝔼_sp___[ℰs___]] := CF /@ 𝔼_sp[ℰs];
CF[𝕌_sp___[Us___]] := CF /@ 𝕌_sp[Us];
```

Logging CCF

In[∘]:=
```
$MaxTime = 0;
$CCFLogFile = OpenWrite[EchoLabel["CCFLogFile is"]@StringJoin[
    "CCFLog-", StringReplace[":" → "-"][DateString["ISODateTime"]], ".m"
   ]];
CCF[ℰ_] := PP_CCF@Module[{ti = TimeUsed[], r},
   r = Together[ℰ];
   If[TimeUsed[] - ti > $MaxTime,
    $MaxTime = TimeUsed[] - ti;
    Write[$CCFLogFile, {Echo@$MaxTime, ℰ}]
   ];
   r]
```

» CCFLogFile is CCFLog-2021-08-17T11-42-18.m

pdf

Variables and their duals:

pdf

In[∘]:=
```
{t*, b*, y*, a*, x*, z*, τ*, β*, η*, α*, ξ*, ξ*} = {τ, β, η, α, ξ, ξ, t, b, y, a, x, z};
(vs_List)* := (v ↦ v*) /@ vs;
(u_i_)* := (u*)_i;
𝓕[u_i_] := 𝓕[u_i] = ToExpression["𝓕" <> ToString[u]]_i
```

pdf

Weights:

pdf

In[∘]:=
```
Clear[Wt];
Evaluate[Wt /@ {y, b, t, a, x, η, β, τ, α, ξ}] = {1, 0, 0, 2, 1, 1, 2, 2, 0, 1};
Wt[u_i_] := Wt[u];
```

pdf

The maximal weight $n, i.e. the *n* of gl(*n*). Initially and for a long while this will not be tested beyond $n == 2.

pdf

In[ ]:=
```
$n = 2;
```

pdf

Upper to lower and lower to Upper:

pdf

In[ ]:=
```
U2l[ℰ_] := ℰ /. {B_i_^p_. :> ⅇ^(-p ℏ b_i), B^p_. :> ⅇ^(-p ℏ b), T_i_^p_. :> ⅇ^(p ℏ t_i), T^p_. :> ⅇ^(p ℏ t), 𝒜_i_^p_. :> ⅇ^(p α_i), 𝒜^p_. :> ⅇ^(p α)};
l2U[ℰ_] := ℰ //. {ⅇ^(c_. b_i_ + d_.) :> B_i^(-c/ℏ) ⅇ^d, ⅇ^(c_. b + d_.) :> B^(-c/ℏ) ⅇ^d, ⅇ^(c_. t_i_ + d_.) :> T_i^(c/ℏ) ⅇ^d, ⅇ^(c_. t + d_.) :> T^(c/ℏ) ⅇ^d,
    ⅇ^(c_. α_i_ + d_.) :> 𝒜_i^c ⅇ^d, ⅇ^(c_. α + d_.) :> 𝒜^c ⅇ^d, ⅇ^𝒳_ :> ⅇ^(Expand@l2U@𝒳)};
l2U[r_Rule] := Module[{U = r〚1〛 /. {b → B, t → T, α → 𝒜}}, U → l2U[U2l[U] /. r]];
AlsoUpper[rs_List] := rs ⋃ (l2U /@ rs);
```

pdf

Derivatives in the presence of exponentiated variables:

pdf

In[ ]:=
```
D_b[f_] := ∂_b f - ℏ B ∂_B f; D_b_i_[f_] := ∂_b_i f - ℏ B_i ∂_B_i f;
D_t[f_] := ∂_t f + ℏ T ∂_T f; D_t_i_[f_] := ∂_t_i f + ℏ T_i ∂_T_i f;
D_α[f_] := ∂_α f + 𝒜 ∂_𝒜 f; D_α_i_[f_] := ∂_α_i f + 𝒜_i ∂_𝒜_i f;
D_v_[f_] := ∂_v f;
```

pdf

𝔼 operations:

pdf

In[ ]:=
```
ℰ_𝔼[$] := Length[ℰ] - 1; 𝔼_[ℰs___][$] := 𝔼[ℰs][$];
ℰ_𝔼[k_Integer] := ℰ〚k + 1〛; 𝔼_[ℰs___][k_Integer] := {ℰs}〚k + 1〛;
𝔼 /: ℰ1_𝔼 ≡ ℰ2_𝔼 := Inner[CF@#1 == CF@#2 &, ℰ1, ℰ2, And];
𝔼_(d1_→r1_)[ℰ1s___] ≡ 𝔼_(d2_→r2_)[ℰ2s___] ^:= (d1 == d2) ∧ (r1 == r2) ∧ (𝔼[ℰ1s] ≡ 𝔼[ℰ2s]);
𝔼 /: ℰ1_𝔼 * ℰ2_𝔼 := 𝔼 @@ Table[CF[ℰ1[kk] + ℰ2[kk]], {kk, 0, Min[ℰ1[$], ℰ2[$]]}];
𝔼_(d1_→r1_)[ℰ1s___] 𝔼_(d2_→r2_)[ℰ2s___] ^:= 𝔼_((d1⋃d2)→(r1⋃r2)) @@ (𝔼[ℰ1s] 𝔼[ℰ2s]);
```

pdf

In[ ]:=
```
𝔼_(d1_→r1_)[ℰ1s___] // 𝔼_(d2_→r2_)[ℰ2s___] := Module[{is = r1 ⋂ d2, lvs},
  lvs = Flatten@Table[{y_$@i, b_$@i, t_$@i, a_$@i, x_$@i}, {i, is}];
  𝔼_((d1⋃Complement[d2,is])→(r2⋃Complement[r1,is])) @@ (Zip_(lvs⋃lvs*)[{(ℱ /@ lvs*).(ℱ /@ lvs), Times[
       𝔼[ℰ1s] /. Table[(v : b | B | t | T | a | x | y)_i → v_$@i, {i, is}],
       𝔼[ℰ2s] /. Table[(v : β | τ | α | 𝒜 | ξ | η)_i → v_$@i, {i, is}]
      ]}])
 ]
```

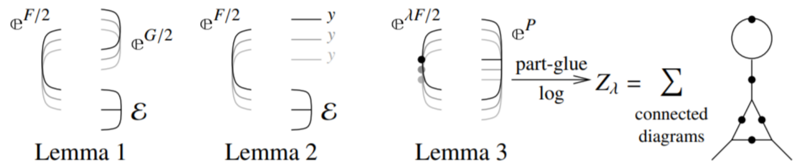
pdf

In[ ]:=
```
Λ2𝔼_(d_→r_)[Λ_] := Module[{k}, 𝔼_(d→r) @@ l2U@Table[SeriesCoefficient[Λ, {ϵ, 0, k}], {k, 0, $k}]];
```

pdf

Zipping! Lemmas 2 and 3 are combined, yet they must be applied first to the middle weight variables and then to the heavy and light variables.

pdf



pdf

Comment. Zip3 of the outer variables must occur after all other operations are completed, because we must allow for gluings of the weight n variables in perturbations with the weight 0 variables in the coefficients of Q.

pdf

```
In[∘]:= Zip_vs_[{𝓕_, 𝓔_}] :=
  {𝓕, 𝓔} // Zip1_vs (*//Zip2_Select[vs, (0<Wt[#]<$n)&]*) // Zip2_Select[vs, (Wt[#]==0∨Wt[#]==$n)&] //
    EZip3_Select[vs, (0<Wt[#]<$n)&] // Zip3_Select[vs, (Wt[#]==0∨Wt[#]==$n)&] // Last;
```

pdf

Getting rid of the quadratic.

Lemma 1. With convergences left to the reader,

$$\left\langle F : \mathcal{E}\,\mathbb{e}^{\frac{1}{2}\sum_{i,j\in B} G_{ij} z_i z_j} \right\rangle_B = \det(1-GF)^{-1/2} \left\langle F(1-GF)^{-1} : \mathcal{E} \right\rangle_B$$

pdf

```
In[∘]:= Zip1_{} = Identity;
Zip1_vs_@{𝓕_, 𝔼[Q_, P___]} := PP_Zip1@Module[{𝓘, F, G, u, v},
   𝓘 = IdentityMatrix@Length@vs;
   F = Table[If[Wt[u] + Wt[v] == $n, ∂_{𝓕[u],𝓕[v]} 𝓕, 0], {u, vs}, {v, vs}];
   G = Table[If[Wt[u] + Wt[v] == $n, ∂_{u,v} Q, 0], {u, vs}, {v, vs}];
   {CF[(𝓕 /@ vs).(F.Inverse[𝓘 - G.F]).(𝓕 /@ vs) / 2],
    𝔼[CF[Q - PowerExpand@Log[Det[𝓘 - G.F]] / 2 - vs.G.vs / 2], P]}
  ]
```

pdf

Getting rid of linear terms.

Lemma 2. $\left\langle F : \mathcal{E}\,\mathbb{e}^{\sum_{i\in B} y_i z_i} \right\rangle_B = \mathbb{e}^{\frac{1}{2}\sum_{i,j\in B} F_{ij} y_i y_j} \left\langle F : \left.\mathcal{E}\right|_{z_B \to z_B + F y_B} \right\rangle_B.$

pdf

```
In[∘]:= Zip2_{} = Identity;
Zip2_vs_@{𝓕_, 𝔼[Q_, P___]} := PP_Zip2@Module[{F, Y, u, v},
   F = Table[If[Wt[u] + Wt[v] == $n, CF[∂_{𝓕[u],𝓕[v]} 𝓕], 0], {u, vs}, {v, vs}];
   Y = Table[∂_v Q, {v, vs}] /. AlsoUpper@Table[v → 0, {v, vs}];
   CF /@ ({𝓕, 𝔼[Q - Y.vs + Y.F.Y / 2, P]} /. AlsoUpper@Thread[vs → vs + F.Y])
  ]
```

pdf

Dealing with Feynman diagrams.

Lemma 3. With an extra variable λ, $Z_\lambda := \log[\lambda F : \mathbb{e}^P]_B$ satisfies and is determined by the following PDE / IVP:

$$Z_0 = P \quad \text{and} \quad \partial_\lambda Z_\lambda = \frac{1}{2}\sum_{i,j\in B} F_{ij}\left(\partial_{z_i}\partial_{z_j} Z_\lambda + (\partial_{z_i} Z_\lambda)(\partial_{z_j} Z_\lambda)\right).$$

Note that the power m of λ is at most $k - 1 + \frac{2k+2}{2} = 2k$. We write $Z_\lambda = \sum Z[m]\,\lambda^m$.

pdf

In[∘]:=
```
Zip3vs_@{𝓕_, ℰ_𝔼} := PPZip3@Module[{F, u, v, Z, $k, kk, jj, $m = 0, m, n},
  $k = Length[ℰ] - 1;
  Do[Z[0, kk] = ℰ⟦kk + 1⟧, {kk, 0, $k}];
  F[u_, v_] := F[u, v] = CF@If[Wt[u] + Wt[v] == $n, ∂𝒻[u],𝒻[v] 𝓕, 0];
  Z[m_, kk_, u_] := Z[m, kk, u] = Du[Z[m, kk]];
  Z[m_, kk_, u_, v_] := Z[m, kk, u, v] = Dv[Z[m, kk, u]];
  For[m = 0, m ≤ 2 $m, ++m, For[kk = 0, kk ≤ $k, ++kk,
    Z[m + 1, kk] = CF@Sum[
        If[F[u, v] === 0, 0, F[u, v]/(2 (m + 1))
          (Z[m, kk, u, v] + Sum[Z[n, jj, u] * Z[m - n, kk - jj, v], {n, 0, m}, {jj, 0, kk}])],
        {u, vs}, {v, vs}];
    If[Z[m + 1, kk] =!= 0, $m = m + 1]
   ]];
  CF /@ ({
      𝓕 - Sum[F[u, v] * 𝒻[u] * 𝒻[v] / 2, {u, vs}, {v, vs}],
      𝔼 @@ Table[Sum[Z[m, kk], {m, 0, $m}], {kk, 0, $k}]
     } /. AlsoUpper@Table[v → 0, {v, vs}])
 ]
```

pdf

Encapsulation.

pdf

In[∘]:=
```
EZip3vs_@{𝓕_, ℰ_𝔼} := PPEZip3@Module[
  {nℰ, n𝓕, rc, ps, rr = {(*release rules*)}, 𝒻vs},
  rc = 0; nℰ = Total[
     CoefficientRules[#, vs] /. (ps_ → c_) ⧴ (AppendTo[rr, cℰ[++rc] → c]; cℰ[rc] (Times @@ vs^ps))
    ] & /@ ℰ;
  rc = 0; 𝒻vs = 𝒻 /@ vs;
  n𝓕 = Total[CoefficientRules[𝓕, 𝒻vs] /.
     (ps_ → c_) ⧴ (AppendTo[rr, c𝓕[++rc] → c]; c𝓕[rc] (Times @@ 𝒻vs^ps))];
  CF[Expand[{n𝓕, nℰ} // Zip3vs] /. rr]
 ]
```

pdf

Exponentials and logarithms as in Exp.nb

pdf

Task. Define $\text{Exp}_m[U:\mathbb{U}_{\{\_\_\}\to\{i\_\}}[\_\_]]$ to compute $e^{\mathbb{O}(U)}$ to order $\epsilon^{\text{Length}@\{U\}-1}$ using the $m_{i,i\to i}$ multiplication, where U is an ϵ-dependent sub-balanced near-docile element, giving the answer in $\mathbb{E}$-form.
Example: $\text{Exp}_{\text{dm},1}[\mathbb{U}_{\{\}\to\{2\}}[b_2\, a_2 + y_2\, x_2, 0]]$ is the exponential of the arrow on strand 2, computed to degree 1.

pdf

```
Exp_m_[U : 𝕌_{is_→{i1_}}[__]] :=
 Module[{λ, μ, k, n, F, f, i, j, lhs, rhs, U1, MI (*multi-index*), mis, mi, yax},
  MI /: Coefficient[ℰ_, MI[p_, n_, q_]] :=
   Coefficient[Coefficient[Coefficient[ℰ, y_i, p], a_i, n], x_i, q];
  yax /: yax^MI[p_,n_,q_] := y_i^p a_i^n x_i^q;
  U1 = U /. (v : (y | b | t | a | x | B | T | 𝒜))_{i1} → v_i;
  F = 𝔼_{is→{i}}[];
  Do[AppendTo[F, 0]; Do[
    mis = Flatten@Table[MI[p, n, q], {p, 0, Min[k + 1, 2 k + 2 - 2 n]}, {q, 0, Min[k + 1, 2 k + 2 - 2 n - p]}];
    F⟦-1⟧ += Sum[f_mi[λ] yax^mi, {mi, mis}];
    lhs = (∂_μ U2l@Last[F (F /. {λ → μ, i → j}) // m_{i,j→i}]) /. μ → 0 /. f_[0] → 0 /.
      Table[f_mi'[0] → Coefficient[U1⟦k + 1⟧, mi], {mi, mis}];
    rhs = ∂_λ U2l@Last[F];
    F = l2U[F /. First@DSolve[
         Table[Coefficient[lhs - rhs, mi] == 0 ∧ f_mi[0] == 0, {mi, mis}], Table[f_mi, {mi, mis}], λ]],
    {n, k + 1, 0, -1}], {k, 0, Length[U1] - 1}];
  CF@l2U[F /. {λ → 1, i → i1}] ]
```

pdf

Logarithms

pdf

Task. Define $\mathrm{Log}_m[\mathcal{E} : \mathbb{E}_{\{\_\_\_\}\to\{i\_\}}[\_\_]]$ to compute $\mathrm{Log}@\mathbb{O}[e^{\mathcal{E}}]$ to order $\epsilon^{\mathrm{Length}@\{U\}-1}$ using the $m_{i,i\to i}$ multiplication, where $\mathcal{E}$ is an ϵ-dependent sub-balanced docile element, giving the answer in $\mathbb{U}$-form.

pdf

```
Log_m_[ℰ : 𝔼_{is_→{i_}}[__]] :=
 Module[{e, k, n, G, c, g, eqn, Sanify, MI (*multi-index*), mis, mi, yax, p, q},
  G = 𝕌_{is→{i}}[c_1 a_i + c_2 x_i y_i]; eqn = U2l[Last[Exp_m[G]] - ℰ⟦1⟧];
  {eqn, G} = CF /@ ({eqn, G} /. First@Solve[Coefficient[eqn, a_i] == 0, c_1]);
  Sanify[{{v_ → s_}}] := v → PowerExpand[Normal[s] /. 𝕔_ → 0];
  G = CF[G /. Sanify@Solve[Coefficient[eqn, x_i y_i] == 0, c_2]];
  G⟦1⟧ += c_0 + c_1 x_i + c_2 y_i; eqn = U2l[Last[Exp_m[G]] - ℰ⟦1⟧];
  {eqn, G} =
   CF /@ ({eqn, G} /. First@Solve[Coefficient[eqn, x_i] == 0 ∧ Coefficient[eqn, y_i] == 0, {c_1, c_2}]);
  G = G /. First@Solve[eqn == 0, c_0];
  MI /: Coefficient[e_, MI[p_, n_, q_]] :=
   Coefficient[Coefficient[Coefficient[e, y_i, p], a_i, n], x_i, q];
  yax /: yax^MI[p_,n_,q_] := y_i^p a_i^n x_i^q;
  Do[
   mis = Flatten@Table[MI[p, n, q], {n, 0, k + 1},
      {p, 0, Min[k + 1, 2 k + 2 - 2 n]}, {q, 0, Min[k + 1, 2 k + 2 - 2 n - p]}];
   AppendTo[G, Sum[g_mi yax^mi, {mi, mis}]];
   eqn = U2l[Last[Exp_m[G]] - ℰ⟦k + 1⟧];
   G = CF[G /. First@Solve[Table[Coefficient[eqn, mi] == 0, {mi, mis}], Table[g_mi, {mi, mis}]]],
   {k, Length[ℰ] - 1}];
  CF[l2U@G]
 ]
```