OBINexus Derivative Tracing System (ODTS)

Minimal Insight Navigation Protocol

This memo formalizes a **bottom-up verification process** to trace the shortest-cost path to a valid derivative solution using ODTS principles.

Step 1: Define Cost Function (Minimal Effort Metric)

- **Action cost**: each algebraic transformation (rule application, substitution, simplification).
- **Goal**: minimize total steps from function definition → validated derivative result.
- **Penalty**: branching into redundant or higher-order unnecessary derivatives.

Step 2: Establish Memory Trace (Memo Tracking)

- Create a **memoized record** of each derivative step.
- Record includes:
- Rule applied (power rule, product rule, etc.).
- Input expression.
- Output expression.
- Prevents re-deriving the same sub-expression.

Step 3: Bottom-Up Verification

1. **Primitive Derivatives**: Start from simplest terms (monomials like x^n).
2. Cost = 1 per rule application.
3. Example: $d(x^3)/dx = 3x^2$.
4. **Aggregate Rules**: Combine primitive results via linearity.
5. Cost = 1 per addition/subtraction.
6. Example: derivative of sum = sum of derivatives.
7. **Cross-Check Symmetry**: Ensure mixed partials match (Schwarz's theorem).
8. Cost = 1 check operation.

Step 4: Path Selection (Minimal Insight Trace)

- At each stage, select the **shortest available path** that:
- Avoids redundant re-derivation.
- Leverages previously memoized steps.
- Leads directly toward the target object (gradient, Hessian, etc.).

This guarantees lowest cumulative cognitive + algebraic cost.

Step 5: Verification Layer

- **Trace Completeness**: Ensure all rules are accounted for.
- **Cross-Partial Consistency**: Confirm Hessian symmetry.
- **Boundary Check**: Identify when higher-order derivatives vanish (exhaustion point).

Example Application: $f(x,y) = x^3 + y^3 - 3xy$

- **Primitive layer**: $d(x^3)/dx = 3x^2$, $d(y^3)/dy = 3y^2$, $d(-3xy)/dx = -3y$.
- **Aggregate**: Gradient assembled from primitive derivatives.
- **Cross-Check**: Mixed partials (-3) consistent.
- **Boundary**: Third derivatives vanish ⇒ trace ends.

Minimal Path Verification Summary

1. Start at primitive rules (lowest cost).
2. Build up gradient from memoized primitives.
3. Construct Hessian with symmetry check.
4. Terminate at exhaustion point.

This sequence defines the **shortest valid derivation path** under ODTS, ensuring correctness while minimizing algebraic effort.