categorical_picture

> (1) Injection → (2) Push-out/Synthesis → (3) Semantic realisation

I first show the abstract categorical picture, then walk a concrete mini-example ("Birds fly / Penguins don't") all the way through.

---

1 Injection: turning α ∧ ¬α into a morphism

Let

- T_0 – the base theory (a set of sentences closed under entailment)
- **α** – a new assertion
- **¬α** – its negation

$$\begin{tikzcd}
T_0 \arrow[r, "\text{add } \alpha"] \arrow[d, "\text{id}"] & T_1 := T_0 \cup \{\alpha\} \\
T_0 \arrow[r] & T_0' := T_0 \cup \{\neg\alpha\} \arrow[u, "\text{add } \neg\alpha"]
\end{tikzcd}$$

Both arrows are *injections* (monos) in **Th**, the paraconsistent-theory category.

Together they realise the formal contradiction; nothing collapses because hom-sets are enriched over Belnap's 4-value lattice .

---

2 Push-out (synthesis): the universal "glue" object T_2

In **Th** every span $T_1 \leftarrow T_0 \rightarrow T_0'$ has a colimit:

```
         i_1
T_1 ────────────────▶ T_2 ◀──────────────── T_0'
 ▲          ▲          ▲
 |          | i_2      |
 |          |          |
 └───────── id ◀───────┴───────── id ◀─────┘
      T_0
```

Concrete construction (your Prop. 4.2 idea).

Introduce a fresh symbol *p* that “re-threads” the conflict:

$$
\begin{aligned}
T_2 := T_0 \\
\quad \cup \{\alpha \to p, \\
\qquad \neg\alpha \to \neg p, \\
\qquad p \lor \neg p\} \qquad & \text{(minimal, adds just one atom)}
\end{aligned}
$$

p is the **new emergent concept**; T_2 is initial among theories receiving both injections while preserving entailment relations.

Mathematically: for any theory **S** and maps $f_1{:}T_1 \to S$, $f_2{:}T_0' \to S$ that agree on T_0, there exists a **unique** $h{:}T_2 \to S$ with $h \circ i_1 = f_1$ and $h \circ i_2 = f_2$.

---

3 Semantic realisation via a functor

F : Th → Sem

Choose **Sem** to be a suitable model category (sets, Kripke frames, graphs…).

Below, I use **Set** of *two-world Kripke models* for clarity.

```
F(T_0)  ← F(T_1)
  ↘        ↘
    colim ───→ F(T_2)       (preserves the push-out)
  ↗        ↗
F(T_0') ←
```

Minimal two-world model that realises T_2

World	p	α	¬α
w_1	T	T	F
w_2	F	F	T

- The link $p \lor \neg p$ ensures every world lands in exactly one context.
- No explosion: both worlds coexist; $\alpha \land \neg\alpha$ holds globally in four-valued sense, yet each world is classically consistent.

Walk-through with a tangible KB fragment

> Thesis:

> Antithesis:

① Injection

T_0 = taxonomy facts (penguin(x) → bird(x)).

Add α on one branch, ¬α on the other.

② Push-out

Introduce context(penguin) ≡ *p*.

Add rules

```
α → p           (flying rule applies only in p-world)
¬α → ¬p           (non-flying rule in ¬p-world)
p ∨ ¬p          (each individual tagged)
```

Result: the KB now **contains a fresh attribute** “penguin-context” that reconciles both flight rules.

③ Semantic image

A data store instantiates two labelled sub-graphs or two rows in a truth-table. Query engines can now answer:

- “Do birds fly (generically)?” – Yes in *p*world.
- “Do penguins fly?” – No in ¬*p*world.

Explainability: system says *“Conflict resolved by creating a context dimension ‘penguin’.”*

Visual summary (one slide-style)

```
┌──────────────┐     α          ┌──────────────┐
│  T₁       │─────────────────────▶│        │
│ T₀ ∪ {α}  │           │  T₂    │
                           │ push-out  │
  ▲ id                 │ adds  p   │
   │                   └──────────────┘
   │                   ▲
id   │                 │ ¬α
   │                   │
┌──────────────┐     ¬α          │
│  T₀'      │────────────────────────┘
│ T₀ ∪ {¬α}  │
└──────────────┘
        F ↓                F ↓
──────────────────────────────────────────────────

────────────
        Semantic functor F
   two-world Kripke model  {w₁,w₂}  with p tagging
```

Key play-out: contradiction **injects**, system **pushes-out** to minimal extension, functor **realises** it as a concrete, queryable structure.

That’s the life-cycle your framework formalises—and why we can mirror it in option portfolios, KB repair, or neural concept-splitting.