

ITEM #188 - From Exact Calling Path to Feasible Calling Path

Large-to-Small Containment, Satisfaction Constraints, and Minimal Evolution Threshold

Conversation: Feasible Calling Path

20251229

Authors: Sizhe Tan & GPT-Obot

ITEM #188 (EN)

From Exact Calling Path to Feasible Calling Path

Large-to-Small Connectors, Satisfaction Constraints, and Minimal Evolution Threshold

1. Background

So far, ACLM Calling Graphs have been restricted to **Exact Calling Paths**:

- Interfaces, states, and semantic slots must match exactly.
- Composition follows a “take exactly what you need” principle.
- This produces minimal redundancy and elegant programs.

However, this model is **structurally fragile**:

- If an exact path $X \rightarrow Y$ does not exist, the system fails outright.
- Higher-level solutions that *do* solve the problem are structurally excluded.

In real engineering systems and evolutionary problem solving, such failure is unacceptable.

2. Core Insight: $X\text{-parent} \rightarrow Y\text{-parent}$ Already Solves $X \rightarrow Y$

Let:

- X be the required input capability/state.
- Y be the required output capability/state.

If there exist:

- X_p (**X-parent**) such that $X \in X_p$
- Y_p (**Y-parent**) such that $Y \in Y_p$
- An **Exact Calling Path** $X_p \rightarrow Y_p$ in the calling graph

Then:

The programming problem $X \rightarrow Y$ is already solved.

The resulting program may contain unused or redundant capabilities, but:

- Functional correctness is guaranteed.
- User requirements are fully satisfied.
- Redundancy is an optimization issue, not a correctness issue.

This defines a **Feasible Calling Path with Large-to-Small Connectors**.

3. Satisfaction as a Hard Constraint

Why Symmetric Cosine Similarity Is Invalid

The traditional cosine similarity:

$$\frac{\text{Large} \cdot \text{Small}}{|\text{Large}| \cdot |\text{Small}|} = \frac{\text{Large} \cdot \text{Small}}{|\text{Large}| \cdot |\text{Small}|}$$

cannot be used when $|\text{Large}| > |\text{Small}|$.

In real systems:

- $|\text{Large}|$ strictly contains additional dimensions.
- Therefore $|\text{Large}| > |\text{Small}|$.

- Even when Small is fully contained in Large, the score can never reach 1.0.

Requiring this condition is mathematically impossible and **structurally counterproductive**.

4. The Correct Metric: Small-Anchored Containment Score

We formally define the **Satisfaction Hard Constraint** as:

ContainmentScore(Large \Rightarrow Small)=Large·Small|Small|
 |Small|=1.0 $\boxed{\text{ContainmentScore}(Large \Rightarrow Small) = \frac{Large \cdot Small}{|Small|} = 1.0}$

Key properties:

- **Asymmetric by design**
- Anchored on **Small (user requirement)**
- Verifies **complete requirement satisfaction**, not similarity

Interpretation:

All requirement dimensions of Small are fully present in Large.

This metric is simple, computable, and semantically exact.

5. Two-Phase Search with Satisfaction Constraint

Phase 1: Parent Frontier Enumeration

For X:

- Enumerate candidate parents using Metric Differential Trees or indexes.
- **Strictly filter:**

ContainmentScore(Xp \Rightarrow X)=1.0 $\text{ContainmentScore}(Xp \Rightarrow X) = 1.0$

For Y:

- Same procedure.

Results:

- **PX**: valid X-parents
 - **PY**: valid Y-parents
-

Phase 2: Exact Path Search

- Search **Exact Calling Paths** from PX to PY.
 - Rank solutions by **Minimal Evolution Threshold**:
 - Parent overhead
 - Path length
 - Estimated runtime cost
-

6. Constraint Relaxation Strategy

If no feasible path exists under `ContainmentScore = 1.0`:

1. Enter a **controlled relaxation loop**
2. Gradually relax the threshold:
 - $1.0 \rightarrow 0.99 \rightarrow 0.97 \rightarrow \dots$
3. Each relaxation round is explicit, traceable, and user-visible

Correctness and trust are preserved.

7. Example: Unaligned AND in Large DNA Domains

In genomic computing:

- **Large**: a massive DNA domain or genome window
- **Small**: a target DNA segment or motif
- Objective: determine whether Small exists in Large **without alignment**

This is an **Unaligned AND / Containment Query**, not a similarity query.

Why symmetric similarity fails

Using symmetric cosine similarity would ask:

“Is a million-base genome similar to a 20-base motif?”

This is biologically meaningless and computationally destructive.

Why Small-anchored containment is the only feasible metric

Using:

$$\frac{Large \cdot Small}{|Small|^2} = 1.0 \implies \frac{Large \cdot Small}{|Small|^2} = 1.0 \implies Large \cdot Small = |Small|^2$$

means:

All structural information of the target DNA segment is present in the domain.

This approach is:

- Directionally correct
- Scale-independent
- Index-friendly
- Proven effective in unaligned DNA search and DBM-DNAStarmap designs

In such domains, this score is **the only feasible and meaningful criterion**.

8. Minimal Evolution Threshold Principle

Among all feasible solutions, the system prefers:

- Smaller parents
- Shorter paths
- Less redundant capability expansion

This defines the **Minimal Evolution Threshold**:

Cross the feasibility boundary with the smallest structural expansion.

9. Architectural Implications

This mechanism upgrades ACLM from:

- **Exact-only aristocratic composition**

to:

- **Survival-first, satisfaction-guaranteed, optimization-later structural intelligence**
-

10. Status

- Feasible Path Search: **Engineering-ready**
 - Trimming / Exactification: **Future ACLM research**
 - Satisfaction Constraint + Relaxation: **Core DBM design rule**
-
-

ITEM #188 (中文)

从 Exact Calling Path 到 Feasible Calling Path

Large-to-Small Connector、满意度硬约束与最小进化门槛

1. 背景

目前 ACLM Calling Graph 严格限定于 **Exact Calling Path**：

- 接口、状态、语义槽位完全精确匹配；
- 程序组合“吃多少，拿多少”；
- 结构优雅、冗余最小。

但这一模式在工程上**极其脆弱**：

- 一旦 $X \rightarrow Y$ 的 exact path 不存在，系统直接失败；
- 即便存在更高层级的可行方案，也被结构性排除。

2. 核心转角： $X_p \rightarrow Y_p$ 已经解决 $X \rightarrow Y$

设：

- X ：所需输入能力/状态；
- Y ：所需输出能力/状态。

若存在：

- X_p (X -parent)，满足 $X \in X_p$
- Y_p (Y -parent)，满足 $Y \in Y_p$
- Calling Graph 中存在 **Exact Path**： $X_p \rightarrow Y_p$

则：

$X \rightarrow Y$ 的编程问题已经被解决。

程序可能包含多余功能，但：

- 正确性成立；
- 用户需求满足；
- 冗余属于优化问题。

3. 满意度硬约束

为什么对称 CosSimilarity 是错误的

传统公式：

$$\frac{Large \cdot Small}{|Large| \cdot |Small|} = \frac{Large \cdot Small}{|Large| \cdot |Small|}$$

在 $Large > Small$ 情况下 永远无法等于 1.0 :

- $Large$ 必然包含额外维度 ;
- 分母被 $Large$ 主导 ;
- 即便 $Small$ 完全被包含 , 也会被判定为失败 。

这是数学上不可达的、工程上反动的检验 。

4. 正确的度量 : Small-Anchored Containment

正式定义满意度硬约束为 :

$ContainmentScore(Large \Rightarrow Small) = \frac{Large \cdot Small}{|Small|}$
 $|Small| = 1.0 \Rightarrow ContainmentScore(Large \Rightarrow Small) = Large$
 $ContainmentScore(Large \Rightarrow Small) = 1.0 \Rightarrow |Small| = Large \cdot Small$

特性 :

- 刻意不对称
- 以 $Small$ (用户需求) 为锚点
- 检验的是需求是否被完整满足

含义 :

$Small$ 的所有需求维度 , 在 $Large$ 中一个不缺 。

5. Two-Phases Search (带满意度约束)

Phase 1 : Parent Frontier 枚举

- 通过差分树/索引枚举 parent ；
- 严格过滤：

$$\text{ContainmentScore}(X_p \Rightarrow X) = 1.0 \text{ } \text{ContainmentScore}(X_p \Rightarrow X) = 1.0$$

得到 PX、PY。

Phase 2：Exact Path 搜索

- 仅在 $PX \rightarrow PY$ 上搜索；
- 按最小进化门槛排序：
 - parent 冗余
 - 路径长度
 - 运行成本

6. 约束松弛机制

若硬约束无解：

- 启动可控松弛：
 - $1.0 \rightarrow 0.99 \rightarrow 0.97 \dots$
- 每一轮可解释、可追溯、可暴露给用户。

7. DNA 示例：大型 DNA Domain 中的 Unaligned AND

在 DNA 计算中：

- Large：巨大 DNA Domain / Genome Window

- Small：目标 DNA 片段
- 目标：判断 Small 是否存在于 Large 中（不对齐）

这是一个 **Unaligned AND / 包含性查询**。

对称相似度的失败

询问：

“一个百万 bp 的序列是否和几十 bp 的片段相似”

在生物学与工程上都是无意义的。

Small-anchored 度量的必然性

$$\frac{Large \cdot Small}{|Small|^2} = 1.0 \quad \frac{Large \cdot Small}{|Small|^2} = 1.0 \quad |Small|^2 \cdot Large \cdot Small = 1.0$$

等价于：

Small 的全部结构信息，在 Domain 中被完整 AND 覆盖。

在该场景中，这个 score 是：

- 唯一可行
- 工程可落地
- 生物学合理

8. 最小进化门槛原则

在所有可行解中，优先选择：

- parent 最小
- 路径最短

- 冗余最少
-

9. 架构意义

ACLM 从：

- 只能贵族式精确对接

升级为：

- 先生存、再瘦身、全程可解释的结构智能系统
-

10. 当前状态

- Feasible Path Search：已可工程化
 - Trimming / Exact 化：远景研究
 - 满意度硬约束 + 松弛：DBM 核心规则
-