

# Toy / Solvable Model Problem

在上次作業的第二題中，我把 20 年後的 AI 能力想像成：

「能閱讀自然語言或 LaTeX 的數學敘述，自動將其形式化，並在大型公理系統（如 ZFC、Hilbert 系統或現代 ITP）中，設計出新的證明策略與輔助引理，最後給出可被機器檢查的完整證明。」

## a 問題設計：命題邏輯中的「鏈結證明」

Toy model 的世界只包含三個命題變數  $p, q, r$ ，語言由  $\wedge, \rightarrow$  構成。

給定一組前提與結論，例如：

- 前提： $p \rightarrow q, q \rightarrow r$
- 目標：證明  $p \rightarrow r$

任務定義為：

在固定的自然演繹系統裡（只允許 **Assumption**、**Modus Ponens** 和  $\rightarrow$ -introduction），找到一條從前提出發、到達目標公式的「最短推理序列」。

輸入可以形式化成：

- 一組前提公式的字串或編碼向量
- 一個目標公式

輸出是一串有序推理步驟，每一步都標記：

- 用到的前提／中間結果
- 使用的推理規則

## b 模型與方法：以 BFS + 規則編碼求解

為了讓這個模型是「solvable」，我先不用大型深度模型，而採用一個可完全掌控的組合式方法：

1. 實作「證明狀態」類別，紀錄目前已知命題集合
2. 定義推理規則（Assumption、Modus Ponens、 $\rightarrow$ -introduction）
3. 對狀態空間做 **廣度優先搜尋（BFS）**：
  - 初始節點只有前提
  - 每一層嘗試所有可能的推理規則，產生新節點

- 一旦某個節點包含目標公式，就回溯得到完整證明路徑

在這個極小系統中，狀態數量與推理深度都可手算確認：

對所有「由長度不超過 3 的鏈結所產生的目標公式」，BFS 在深度  $\leq 4$  就能找到證明，因為所有可能子式都已被枚舉。

---

## c 實作結果與觀察

---

我讓程式自動產生多組形如  $(p_1 \rightarrow p_2, p_2 \rightarrow p_3, \dots)$  的前提與對應結論，並用 BFS 尋找證明。

所有測試樣本中，搜尋皆能在固定深度內找到可驗證證明。

例如下列自動生成的證明：

1. 假設  $p$
2. 由  $p \rightarrow q$  與  $p$  推  $q$
3. 由  $q \rightarrow r$  與  $q$  推  $r$
4. 由「假設  $p$  推  $r$ 」推出  $p \rightarrow r$  ( $\rightarrow$ -introduction)

這代表即使不使用神經網路，只用簡單搜尋也能完整處理 toy problem。

未來若加入 machine learning，可以把 BFS 的完整證明路徑當成「教師」，生成大量（前提，結論，證明軌跡）的資料，訓練 policy 或 value network，逐步過渡到 AlphaZero 式的「神經 + 搜尋」架構。

---

## d 與終極目標的關聯

---

這個玩具模型雖然小，但它逼我回答兩個根本問題：

1. 「證明」是什麼型態的物件？  
→ 在這裡是「在推理規則圖上的一條路徑」
2. AI 要學的是什麼？  
→ 不是記住結論，而是「在狀態空間中選擇下一步推理的策略」

因此，這個 toy / solvable model 是我 20 年後願景的縮影：

它小到現在就能解，但結構上與最終目標一致。

接下來最自然的延伸，就是：

- 擴展語言（加入量詞、等號）
- 增加公理庫
- 用學習方法取代人工 BFS 搜尋策略

這些都將讓我逐步逼近「能自動發現並完成嚴格數學證明的 AI 助理」。

