Assignment 11

一、摘要

要讓 AI 在 20 年後達到「能自主提出新的數學定理並給出可正式驗證的證明」的能力，最核心的能力有三項：

1. **從觀測資料中發現可能的結構（Conjecture）**
2. **把結構轉為嚴格的、無歧義的數學命題（Formal Theorem）**
3. **以 Proof Assistant（例如 Lean）完成正式可檢查的證明（Formal Proof）**

但在目前，全自動的數學定理發現與正式證明仍不可能，所以我們設計一個「目前真正可行」的小範圍：

在線性遞迴整數數列的世界中，讓 AI 自動提出定理並給出形式化的表述

因為線性遞迴本身結構簡單、可由資料推回遞迴式，而 closed form 也可以用特徵方程求得，整體流程能完整涵蓋「猜 → 定理 → 證明」的雛型。

二、問題設定

給定未知規律的整數數列前四項：a_0, a_1, a_2, a_3

AI 的任務是：

1. **從資料推測遞迴公式：**$a_n = c_1 a_{n-1} + c_2 a_{n-2}$
2. **求出其 closed from，並寫成一個正式的定理**
3. **利用 Lean 4 將定理形式化，使其成為可驗證的程式化命題**

三、示範：由資料推回遞迴，再到一般項

考慮數列：$a_0 = 2, a_1 = 5, a_2 = 12, a_3 = 29$

Step 1 – AI 自動猜測遞迴式（Conjecture）

假設數列滿足

$$a_n = c_1 a_{n-1} + c_2 a_{n-2}$$

利用前三筆資料求解：

$$\begin{cases} 12 = c_1 \cdot 5 + c_2 \cdot 2 \\ 29 = c_1 \cdot 12 + c_2 \cdot 5 \end{cases}$$

解得：

$$c_1 = 2, c_2 = 1$$

因此可得到一個合理的 Conjecture：

$$a_n = 2a_{n-1} + a_{n-2}$$

Step 2 – AI 產生一般項並寫成正式定理（Theorem）

解特徵方程：

$$x^2 - 2x - 1 = 0$$

解得根：

$$r_1 = 1 - \sqrt{2}, r_2 = 1 + \sqrt{2}$$

因此一般項形式為：

$$a_n = \alpha r_1^n + \beta r_2^n$$

利用初始條件 $a_0 = 2, a_1 = 5$，可得：

$$\alpha = \frac{(1+\sqrt{2}) \cdot 2 - 5}{2\sqrt{2}} = 1 - \frac{3\sqrt{2}}{4}, \qquad \beta = \frac{5 - (1-\sqrt{2}) \cdot 2}{2\sqrt{2}} = 1 + \frac{3\sqrt{2}}{4}$$

正式定理（Formal Theorem）

Theorem.

Let sequence $\{a_n\}$ satisfy $a_0 = 2, a_1 = 5, a_n = 2a_{n-1} + a_{n-2}$. Then for all n, $a_n = \alpha\left(1+\sqrt{2}\right)^n + \beta\left(1-\sqrt{2}\right)^n$, where $\alpha = 1 - \frac{3\sqrt{2}}{4}$, $\beta = 1 + \frac{3\sqrt{2}}{4}$.

四、Python 程式碼：由資料推測遞迴與一般項

```python
import sympy as sp

# Given sequence data
data = [2, 5, 12, 29]

# Step 1: Guess recurrence a_n = c1*a_(n-1) + c2*a_(n-2)
c1, c2 = sp.symbols("c1 c2", real=True)
eqs = []

for n in range(2, len(data)):
    eqs.append(sp.Eq(data[n], c1*data[n-1] + c2*data[n-2]))

sol = sp.solve(eqs, (c1, c2))
print("Reccurence:", sol)

# Step 2: Solve characteristic equation
c1_val, c2_val = sol[c1], sol[c2]
x = sp.symbols("x")
char_eq = sp.Eq(x**2 - c1_val*x - c2_val, 0)
roots = sp.solve(char_eq, x)
print("Roots:", roots)

# Step 3: General form
r1, r2 = roots
α,β = sp.symbols("α β")
n = sp.symbols("n")

general = α*r1**n + β*r2**n

# Solve α, β from initial conditions
sol_ab = sp.solve([
    sp.Eq(general.subs(n, 0), data[0]),
    sp.Eq(general.subs(n, 1), data[1])
], [α, β])

print("Closed-form a_n = α*r1^n + β*r2^n")
print(sol_ab)
```

執行後求得 $c_1, c_2, r_1, r_2, \alpha, \beta$：

```
Reccurence: {c1: 2, c2: 1}
Roots: [1 - sqrt(2), 1 + sqrt(2)]
Closed-form a_n = α*r1^n + β*r2^n
{α: 1 - 3*sqrt(2)/4, β: 1 + 3*sqrt(2)/4}
```

也就是完成「猜遞迴 ＋ 求 closed form」的全部步驟。

五、Lean 4：定理的形式化表達

以下程式碼可以在 Lean Playground 執行：

```
import Mathlib.Data.Real.Basic
import Mathlib.Analysis.SpecialFunctions.Pow.Real

theorem recur_closed_form
  (a : ℕ → ℝ)
  (h0 : a 0 = 2)
  (h1 : a 1 = 5)
  (hrec : ∀ n ≥ 2, a n = 2 * a (n-1) + a (n-2)) :
  ∀ n,
  a n
    = ( (5 - (1 - Real.sqrt 2) * 2) / (2 * Real.sqrt 2) ) * (1 + Real.sqrt 2)^n
    + ( ((1 + Real.sqrt 2) * 2 - 5) / (2 * Real.sqrt 2) ) * (1 - Real.sqrt 2)^n := by
  -- Proof idea:
  -- 1. Solve characteristic equation x^2 - 2x - 1 = 0.
  -- 2. Use general theory of second-order linear recurrences.
  -- 3. Determine α, β from initial values.
  -- 4. Conclude the closed form by induction.
  sorry
```

雖然證明以 sorry 代替，但這表示：

- Lean 已成功解析並接受此 Theorem
- 命題本身語法與類型正確
- 只需補上證明即可完成形式化的驗證

這就是「AI 產生可形式驗證定理」的核心能力。

六、結論

未來目標能力	在簡化模型中如何實現
AI 自動產生 Conjecture	從資料推得 $a_n = 2a_{n-1} + a_{n-2}$
AI 形式化命題	推出 closed form 並寫成 Theorem
AI 產生可檢查 Proof	Lean theorem + proof skeleton
AI 與程式結合	Python 找規律，Lean 檢查定理

這個簡化模型雖然小，但具備完整的數學推理流程，是邁向「AI 可自主作數學研究」的第一步。