

Report – Canonical Knowledge

- Show your autograder result for each question in the report.
- Describe your algorithm for each question in the report.
- Q1 – Logic Warm-up

```
Question q1
=====

*** PASS: test_cases/q1/correctSentence1.test
*** PASS
*** PASS: test_cases/q1/correctSentence2.test
*** PASS
*** PASS: test_cases/q1/correctSentence3.test
*** PASS
*** PASS: test_cases/q1/entails.test
*** PASS
*** PASS: test_cases/q1/entailsLong.test
*** PASS
*** PASS: test_cases/q1/findModelSentence1.test
*** PASS
*** PASS: test_cases/q1/findModelSentence2.test
*** PASS
*** PASS: test_cases/q1/findModelSentence3.test
*** PASS
a.__dict__ is: {'op': 'A', 'args': ()}
*** PASS: test_cases/q1/findModelUnderstandingCheck.test
*** PASS
*** PASS: test_cases/q1/plTrueInverse.test
*** PASS

### Question q1: 10/10 ###

Finished at 17:16:59

Provisional grades
=====
Question q1: 10/10
-----
Total: 10/10
```

在 `sentence1()`, `sentence2()` 中，使用基本的邏輯運算符來構造公式，描述各種條件和結論的邏輯關係，沒涉及到什麼演算法。

在 `sentence3()` 中，使用命題符號來描述 Pacman 的生存狀態。

`findModel` 在尋找一個滿足給定邏輯表達式的模型，若存在這樣的模型則返回 `True`. Otherwise, return `False`.

`Entails` 檢查一個命題是否蘊含 (`entails`) 另一個命題，實質上是檢查在前提為真的情況下，結論是否必然為真。

`plTrueInverse` 用於評估在給定真值分配下，某個表達式的否定形式是否為真。

- Q2 – Logic Workout

```
Question q2
=====

*** PASS: test_cases/q2/atLeastOne.test
*** PASS
*** PASS: test_cases/q2/atLeastOneCNF.test
*** PASS
*** PASS: test_cases/q2/atLeastOneEff.test
*** PASS
*** PASS: test_cases/q2/atMostOne.test
*** PASS
*** PASS: test_cases/q2/atMostOneCNF.test
*** PASS
*** PASS: test_cases/q2/atMostOneEff.test
*** PASS
*** PASS: test_cases/q2/exactlyOne.test
*** PASS
*** PASS: test_cases/q2/exactlyOneCNF.test
*** PASS
*** PASS: test_cases/q2/exactlyOneEff.test
*** PASS

### Question q2: 10/10 ###

Finished at 17:59:41

Provisional grades
=====
Question q2: 10/10
-----
Total: 10/10
```

atLeastOne(): 此函數接受一個表達式列表，並返回一個邏輯表達式，該表達式在 cnf 中表示至少有一個給定的字面量為真。這是透過使用邏輯或運算符將所有字面量聯合起來實現的。

atMostOne(): 此函數生成一個表達式，以 cnf 形式表達最多只有一個給定列表中的字面量為真。透過對每一對字面量創建一個子句實現，每個子句都聲明這兩個字面量不能同時為真 ($\neg A \vee \neg B$)。然後，這些子句透過邏輯與運算符連接起來形成最終的表達式。

exactlyOne(): 此函數結合了前 2 個函數的邏輯，生成一個表達式，表達恰好有一個字面量為真的邏輯。這是通過將 atLeastOne, atMostOne 生成的表達式進行邏輯與操作來實現的，從而確保字面量列表中僅有一個字面量為真。

- Q3 – Pacphysics and Satisfiability

```
Question q3
=====

*** Testing checkLocationSatisfiability
[LogicAgent] using problem type LocMapProblem
Ending game
Average Score: 0.0
Scores:        0.0
Win Rate:      0/1 (0.00)
Record:        Loss
*** PASS: test_cases/q3/location_satisfiability1.test
*** Testing checkLocationSatisfiability
[LogicAgent] using problem type LocMapProblem
Ending game
Average Score: 0.0
Scores:        0.0
Win Rate:      0/1 (0.00)
Record:        Loss
*** PASS: test_cases/q3/location_satisfiability2.test
*** Testing pacphysicsAxioms
*** PASS: test_cases/q3/pacphysics1.test
*** Testing pacphysicsAxioms
*** PASS: test_cases/q3/pacphysics2.test
*** PASS: test_cases/q3/pacphysics_transition.test
***      PASS

#### Question q3: 10/10 ####

Finished at 18:06:42

Provisional grades
=====
Question q3: 10/10
-----
Total: 10/10
```

`pacmanSuccessorAxiomSingle()`: 此函數建立了在給定時間和位置上 Pacman 的繼承狀態公理。這個公理表達了 Pacman 當前位置(x, y)在時間 t 是由前一時間 t-1 透過有效動作達成的結果。代碼首先確認了從可能的四個方向（南、北、東、西）中哪些是無障礙的，然後為每個有效方向創建一個表示 Pacman 從該方向來的表達式。這些表達式透過邏輯或操作聯合起來，與當前位置表達式組成等價關係，形成完整的公理表達式。

`SLAMSuccessorAxiomSingle()`: 這個函數類似於上一個，但它考慮了 Pacman 可能沒有移動的情況。函數首先對可能的移動方向建立表達式，並引入了關於移動失敗（例如撞牆）的額外條件。這些額外條件使用輔助表達式來更精確地描述可能的牆壁和行動組合。最後，這些條件和輔助表達式一起整合成一個包含 Pacman 位置和行動的綜合邏

輯表達式。

pacphysicsAxioms(): 這個函數為給定的時間步 t 建立了一系列的物理和感測公理。它確保 Pacman 不會出現在牆壁位置，並且在每一時間步中只能在一個位置並採取一個行動。此外，如果提供了感測器模型和後繼公理生成函數，它也會集成這些模型的結果。此函數透過聯合所有這些條件來形成一個包羅萬象的邏輯表達式。

● Q4 – Path Planning with Logic

```
### Question q4: 10/10 ###  
  
Finished at 18:42:33  
  
Provisional grades  
=====br/>Question q4: 10/10  
=====br/>Total: 10/10
```

這個函數的核心代碼設計用於解決定位規劃問題，即尋找一系列行動，使 Pacman 從起始位置移動到目標位置。下面將進行整體的功能描述：

positionLogicPlan(): 接收一個位置規劃問題的實例，並返回一個導致目標的行動列表。函數中的算法按時間步逐漸擴充知識庫 (Knowledge Base, KB)，並在每一時間步嘗試查找達到目標的模型。

演算法步驟如下：

1. 初始化和設定：

- ✓ 從問題實例中提取牆壁、寬高度、起始位置和目標位置。
- ✓ 創建可能的坐標列表（排除牆壁位置）和行動列表（北、東、南、西）。
- ✓ 初始化 Knowledge Base，並將 Pacman 的起始位置作為第一條 knowledge 加入。

2. 遞歸時間步的處理：

- ✓ 遍歷至多 50 個時間步，每個時間步中，使用以下步驟擴充知識庫：
 - 確保每個時間步只執行一個行動和 Pacman 僅在一個無牆位置。
 - 若不是第一時間步，則根據可能的位置和牆壁情況，添加 Pacman 的成功狀態公理。
- ✓ 在每個時間步嘗試檢查達到目標位置的條件是否成立。

3. 檢查並返回結果：

- ✓ 在每個時間步結束時，檢查是否存在一個模型滿足 Pacman 在目標位置的條件。
- ✓ 若找到模型，則從模型中提取行動序列並返回。
- ✓ 若在 50 個時間步內未找到路徑，則拋出異常。

● Q5 – Eating All the Food

```
### Question q5: 10/10 ###  
  
Finished at 18:55:47  
  
Provisional grades  
=====
```

Question q5: 10/10

Total: 10/10

這個函數的核心代碼旨在解決食物規劃問題，即找出一系列動作使 Pacman 吃掉所有的食物。函數採用逐步擴充知識庫和每一時間步詢問模型的策略。下面是該函數的詳細說明：

`foodLogicPlan()`: 接受一個食物規劃問題的實例，並返回一個行動序列，以便 Pacman 吃掉所有的食物。與前一個問題類似，這個函數不使用 `pacphysicsAxioms`，而是逐步構建知識庫，試圖在每個時間步找到達成目標的行動序列。

算法步驟

1. 初始化和設定：
 - ✓ 從問題實例中獲取牆壁位置、地圖的寬度和高度、起始位置和食物位置。
 - ✓ 創建可能的坐標列表（排除牆壁位置）和行動列表。
 - ✓ 初始化 Knowledge Base，並將 Pacman 的起始位置和所有食物的初始位置加入 KB。
2. 遞歸時間步的處理：
 - ✓ 遍歷至多 50 個時間步，每個時間步中，使用以下步驟擴充知識庫：
 - 確保每個時間步只執行一個行動，並且 Pacman 僅在一個無牆位置。
 - 若不是第一時間步，則添加 Pacman 的移動繼承狀態公理。
 - 對每個食物位置添加條件，確保 Pacman 經過後，該食物在下一時間步消失。
 - ✓ 在每個時間步結束時，檢查所有食物是否都被吃掉。
3. 檢查並返回結果：
 - ✓ 在每個時間步檢查是否存在一個模型使得所有食物位置在下一時間步都為空。
 - ✓ 若找到模型，則從模型中提取行動序列並返回。
 - ✓ 若在 50 個時間步內未找到合適的行動序列，則拋出異常。

Questions According to Lecture

- Q1. According to AI Weekly in the lecture, some experts and scholars such as Karl Friston and Yann LeCun believe: "You can't get to AGI with LLMs."

Nowadays, the prospects of LLM are so optimistic. Why do you think these experts have such ideas? Please elaborate on your views.

他們的看法可能基於幾個關鍵的考量：

1. 模型的局限性：

目前的大型語言模型，如 GPT，主要基於預測下一個詞的機制。這種模式雖然在文本生成等應用上表現出色，但這種學習方式並不足以涵蓋所有人類智能的方面。例如，這些模型缺乏自我意識、情感理解以及能夠像人類那樣進行抽象思維的能力，也就是現在的學習方式會對 AI 系統有一個很大的局限

2. 技術的進步假設：

一些專家認為，即使是最先進的 LLM，也只是數據處理和模式識別工具。這些模型的操作依賴於大量數據和計算力，但它們在理解複雜概念或進行創造性思考方面的能力有限。因此，達到與人類相似的認知靈活性和適應性可能需要新的技術突破。

3. 認知模型的差異：

從認知科學的角度看，人類的思維不僅僅是單純的信息處理；我們的大腦如何整合感覺輸入、情緒、記憶和經驗，形成一個連貫的「自我」，這在目前的 LLM 中尚未得到很好的模擬。AGI 的實現需要模型能夠模仿這種複雜的認知過程。

總而言之，Friston 和 LeCun 等專家對於用現有的 LLM 實現 AGI 持懷疑態度，這反映了他們對目前技術的局限性和未來發展方向的深思熟慮。他們的看法強調了繼續探索 AI 發展的複雜性和多維度的重要性，這種批判性思維某種程度來說也是科技進步必不可少的重要驅動力。

- Q2. According to the paper "CLIP-Event: Connecting Text and Images with Event Structures", in CVPR 2022, after the process of generating the event-centric structured data, how does this work implement contrastive learning? Specifically, how does this work choose the positive and negative samples for contrastive learning?

Contrastive learning 是透過利用文本信息提取技術獲得的事件結構化知識來進行的。其 implement 包括利用多種提示函數來生成困難的負面描述，並設計了基於最優傳輸的事件圖對齊損失。正面樣本通常是事件描述與圖像內容相匹配的結果，而負面樣本則是透過操縱事件結構來創造，例如改變論元角色的順序或選擇視覺特徵與主要事件類型相混淆的事件類型。這種方法強化了模型學習識別正確的事件描述與不匹配的描述之間的區別。

- Q3. Referring to the paper, what is the main problem with the current description of the VLM pre-training process? Please describe the steps to generate the structured graph-based data in this work.

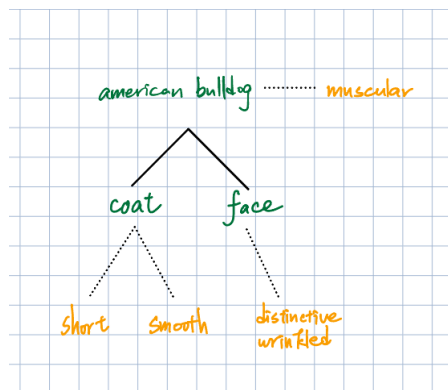
在這篇論文中指出，現有的視覺語言模型（VLM）的預訓練過程主要問題在於，它們主要關注於圖像中的對象與文本中的實體之間的對齊，卻往往忽略了事件結構及其論元角色的複雜性。此工作生成結構化圖形數據的步驟主要包括使用文本信息提取技術來自動從字幕中抽取事件和論元的結構化知識，並將這些信息轉換成圖形結構，以便模型能夠更準確地理解和表示跨媒介的事件知識。這一過程增強了模型對於事件和其參與者角色之間關係的理解，從而優化了事件圖的構建和學習。

- Q4. Please select one of the datasets provided in this work and visualize two categories.

- Path to the gpt-generated data: ./data/gpt_data
- You must show the corresponding description and two graph-based structured data components for each category as shown on the right.

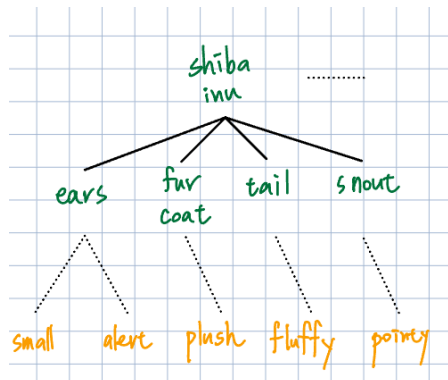
"american bulldog":

"A pet American bulldog is a muscular, medium to large sized dog, with a short, smooth coat and distinctive wrinkled face.",



"shiba inu":

"Shiba Inus are small, agile dogs with thick fur and pointed ears. They typically have a stocky build and come in various shades of tan and red.",



- Q5. Based on current VLM auxiliary data improvement methods, such as the event-centric structure data in the lecture and the structured linguistic knowledge in this paper, what other deep semantic knowledge do you think humans possess that can be provided to VLM for learning?

人類擁有豐富的深層語義知識，對於視覺語言模型（VLM）的學習有很大的幫助。除了事件中心結構數據和結構化語言知識之外，其他可能提供給 VLM 學習的深層語義知識包括：

1. 情感語境理解：如從圖像或文字中辨識情感和情緒的細微差異。
2. 常識推理：包括世界知識、因果關係、目的和意圖理解等。
3. 文化和社會背景知識：如特定的文化符號、習俗和社會規範的識別。
4. 時空連貫性：理解事件在時間和空間上的連續性和進程。
5. 抽象和隱喻理解：識別和理解語言和視覺中的隱喻和抽象概念。

將這些深層語義知識融入 VLM，可以進一步提升模型對複雜人類溝通方式的理解。