**AI Planning**

Our agent performs planning to move cargoes between cities. The agent can load a cargo at a city into an empty cargo space in an airplane that is at the same city. The agent can also unload a cargo from an airplane. The agent can fly an airplane from a city to another city. The airplanes *Plane1* and *Plane2* are at *Melbourne* and the airplane *Plane3* is at *Sydney*. *Plane1* has two cargo spaces *CS11* and *CS12*. *Plane2* has three cargo spaces *CS21*, *CS22* and *CS23*. *Plane3* has two cargo spaces *CS31* and *CS32*. Cargo *C1* is currently occupying cargo space

*CS12* in *Plane1*. Cargo spaces *CS11*, *CS21*, *CS22*, *CS23*, *CS31* and *CS32* are currently empty. Cargoes *C2* and *C3* are currently at *Melbourne*. Cargoes *C4* and *C5* are currently at *Sydney*. The goal is to get the cargoes *C1*, *C2*, and *C3* to *Sydney* and to get the cargoes *C4* and *C5* to *Melbourne*.

1. Write down the initial state description and the agent’s goals.

**Initial State:**

At(Plane1, Melbourne)

At(Plane2, Melbourne)

At(Plane3, Sydney)

In(C1, CS12, Plane1)

At(C2, Melbourne)

At(C3, Melbourne)

At(C4, Sydney)

At(C5, Sydney)

Empty(CS11)

Empty(CS21)

Empty(CS22)

Empty(CS23)

Empty(CS31)

Empty(CS32)

**Goal:**

At(C1, Sydney)

At(C2, Sydney)

At(C3, Sydney)

At(C4, Melbourne)

At(C5, Melbourne)

1. Write down STRIPS-style definitions of the three actions.

**Load(Cargo, CargoSpace, Plane, City):**

* Preconditions:
  + At(Cargo, City)
  + At(Plane, City)
  + Empty(CargoSpace)
* Effects:
  + In(Cargo, CargoSpace, Plane)
  + ¬At(Cargo, City)
  + ¬Empty(CargoSpace)

**Unload(Cargo, CargoSpace, Plane, City):**

* Preconditions:
  + In(Cargo, CargoSpace, Plane)
  + At(Plane, City)
* Effects:
  + At(Cargo, City)
  + Empty(CargoSpace)
  + ¬In(Cargo, CargoSpace, Plane)

**Fly(Plane, FromCity, ToCity):**

* Preconditions:
  + At(Plane, FromCity)
* Effects:
  + At(Plane, ToCity)
  + ¬At(Plane, FromCity)

1. Write down a consistent partial-order plan (POP) with no open preconditions for this problem.

* Unload(C1, CS12, Plane1, Sydney)
* Unload(C2, CS11, Plane1, Sydney)
* Unload(C3, CS21, Plane2, Sydney)
* Load(C4, CS31, Plane3, Sydney)
* Load(C5, CS32, Plane3, Sydney)
* Fly(Plane3, Sydney, Melbourne)
* Unload(C4, CS31, Plane3, Melbourne)
* Unload(C5, CS32, Plane3, Melbourne)

# Uncertain reasoning

Kangaroo Electronics is an electronics manufacturer that uses an AI system FPD to detect faulty products. The FPD system classifies a product into one of two bags: Good and Bad. When a faulty product is examined by FPD, it is classified as Bad by FPD with a probability of 0.98.

When a non-faulty product is examined by FPD, it is classified as Bad by FPD with a probability of 0.01. Statistics from Kangaroo Electronics shows that, on average, there is 1 in 200 products is faulty.

**Given:**

* Probability of a product being faulty: 𝑃(𝐹)=1200=0.005*P*(*F*)=2001​=0.005
* Probability of a product being non-faulty: 𝑃(𝑁𝐹)=1−𝑃(𝐹)=0.995*P*(*NF*)=1−*P*(*F*)=0.995
* Probability of FPD classifying a faulty product as Bad: 𝑃(𝐵∣𝐹)=0.98*P*(*B*∣*F*)=0.98
* Probability of FPD classifying a non-faulty product as Bad: 𝑃(𝐵∣𝑁𝐹)=0.01*P*(*B*∣*NF*)=0.01

1. What is the probability that the next product is classified as Bad by FPD?

Using the law of total probability:

*P*(*B*)=*P*(*B*∣*F*)⋅*P*(*F*)+*P*(*B*∣*NF*)⋅*P*(*NF*)

*P*(*B*)=(0.98⋅0.005)+(0.01⋅0.995)

𝑃(𝐵)=0.0049+0.

𝑃(𝐵)=0.01485*P*(*B*)=0.01485

1. What is the probability that the next product is both faulty and classified as Bad by FPD?

P(F∩B)=P(B∣F)⋅P(F)

P(F∩B)=0.98⋅0.005

𝑃(𝐹∩𝐵)=0.0049

1. What is the probability that the next product is non-faulty and classified as Bad by FPD?

P(NF∩B)=P(B∣NF)⋅P(NF)

𝑃(𝑁𝐹∩𝐵)=0.01⋅0.995

P(NF∩B)=0.00995

1. A product is classified as Bad by FPD, what is the probability that it is actually faulty?

Using Bayes' theorem:

𝑃(𝐹∣𝐵)= [P(B∣F)⋅P(F)​] / P(B)

𝑃(𝐹∣𝐵)= (0.98⋅0.005) / 0.01485

𝑃(𝐹∣𝐵)= 0.0049 / 0.01485

P (F∣B)≈ 0.33

**Machine Learning:**

After training your linear regression model, you observe a training error of 10% but a test error of 45%. What can you infer about this linear regression model?

The observed training error of 10% and a test error of 45% suggest that the linear regression model is overfitting. Overfitting occurs when the model captures the noise in the training data rather than the underlying pattern, leading to poor generalization to unseen data (test set). This is evidenced by the significantly higher error rate on the test data compared to the training data. To mitigate this, techniques such as regularization, cross-validation, or increasing the training data may be employed.