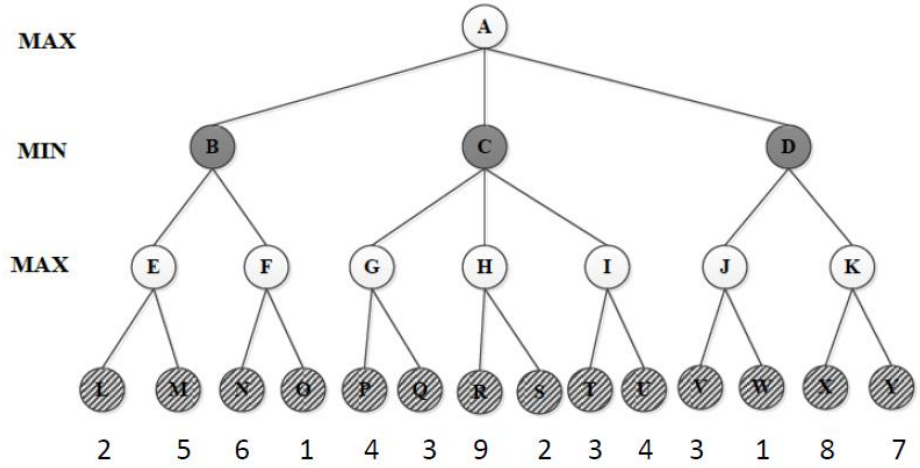
Nama : Kadek Fajar Pramartha Yasodana

NRP : 5025231185

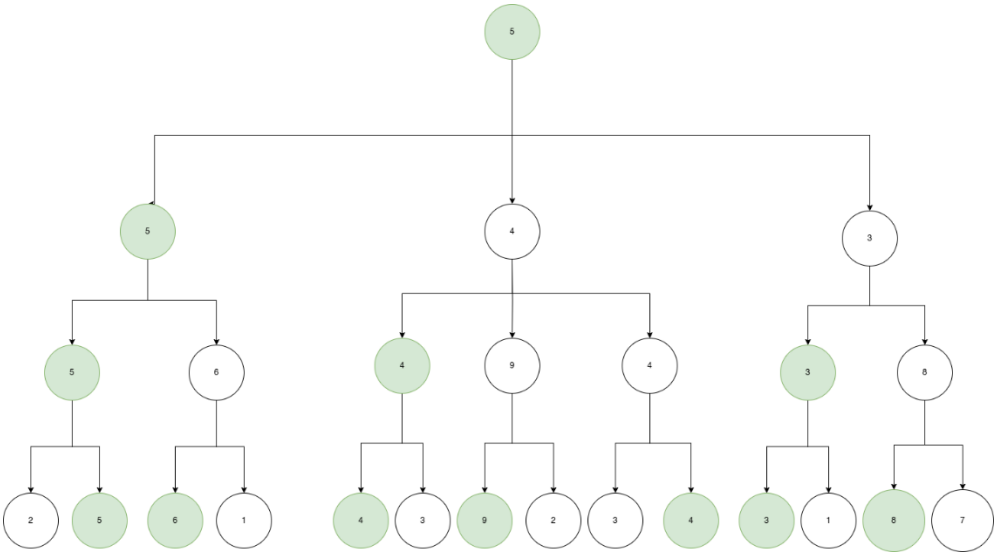
Kelas : KKA D

Tugas Individu

1. Diketahui sebuah adversial tree sebagai berikut:

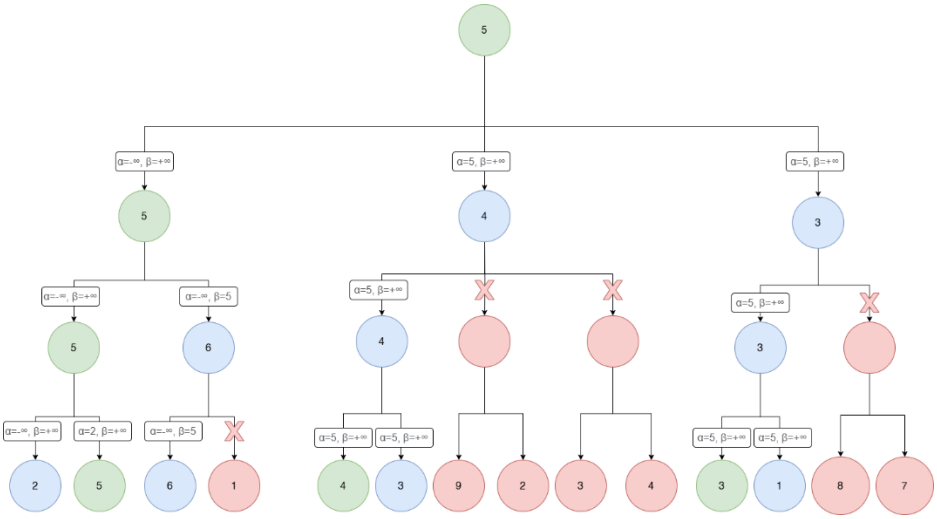


   a. Tentukan nilai maksimum dan minimum pada setiap level menggunakan algoritma Minimax.



   Untuk menentukan nilai maximum/minimum sesuai kriteria pada level kita tinggal perlu melakukan perbandingan pada setiap node.
   Jadi Didapatkan
   Pada level 0, Nilai Maximumnya adalah A=5
   Pada level 1, Nilai Minimumnya adalah B=5, C=4, dan D=3
   Pada level 2, Nilai Maximumnya adalah E=5, F=6, G=4, H=9, I=4, J=3, K=8

   b. Tentukan nilai maksimum dan minimum pada setiap level menggunakan algoritma Alpha-Beta Prunning serta informasi prunning node pada saat level maksimum maupun minimum.



Untuk mendapatkan nilai minimum/maximum menggunakan algorithma alpha beta pruning, terdapat dua buah variable yang akan digunakan untuk melakukan checking demi mengurangi iterasi untuk mendapatkan hasil akhir root. Dengan alpha adalah nilai terbaik untuk maximizer sekarang dan dari root, dan beta adalah nilai terbaik untuk minimizer sekarang dan dari root.
Jadi Didapatkan
Pada level 0, Nilai Maximumnya adalah A=5
Pada level 1, Nilai Minimumnya adalah B=5, C=4, dan D=3
Pada level 2, Nilai Maximumnya adalah E=5, F=6, G=4, H=Pruned, I=Pruned, J=3, K=Pruned

2. You are in charge of scheduling for computer science classes that meet Mondays, Wednesdays and Fridays. There are 5 classes that meet on these days and 3 professors who will be teaching these classes. You are constrained by the fact that each professor can only teach one class at a time.

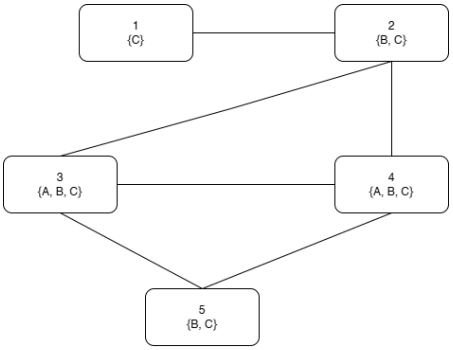
   The classes are:

   • Class 1 - Intro to Programming: meets from 8:00-9:00am
   • Class 2 - Intro to Artificial Intelligence: meets from 8:30-9:30am
   • Class 3 - Natural Language Processing: meets from 9:00-10:00am
   • Class 4 - Computer Vision: meets from 9:00-10:00am
   • Class 5 - Machine Learning: meets from 9:30-10:30am

   The professors are:

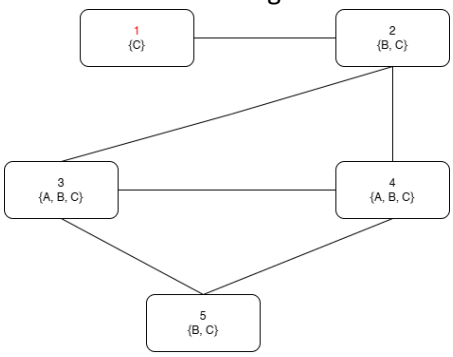
   • Professor A, who is available to teach Classes 3 and 4.
   • Professor B, who is available to teach Classes 2, 3, 4, and 5.
   • Professor C, who is available to teach Classes 1, 2, 3, 4, 5.

   a. Formulate this problem as a CSP problem

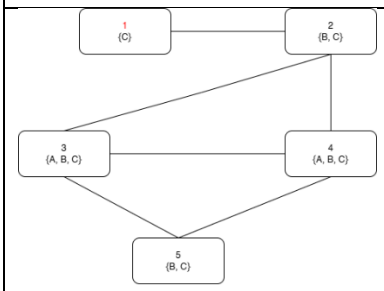
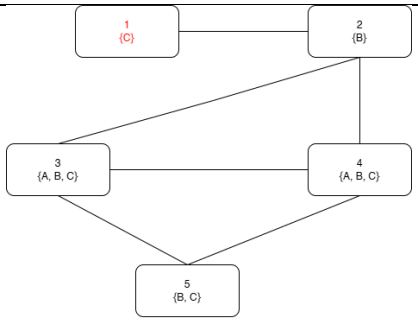


b. Solve this problem using CSP

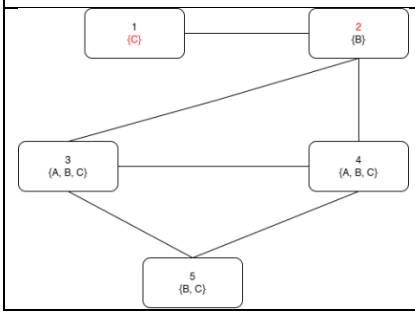
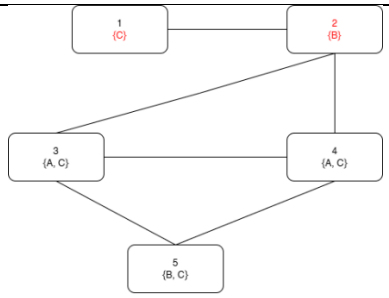
1) Start Forward Checking



2) Choose C

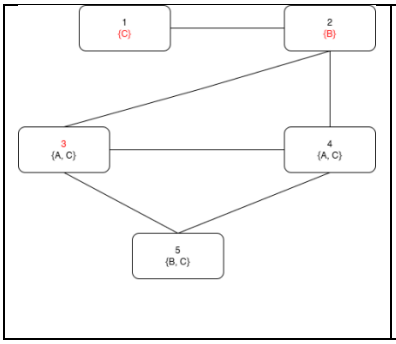
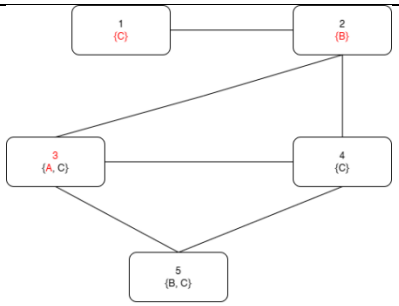
Now	Choose	Next Pos
		2

3) Choose B

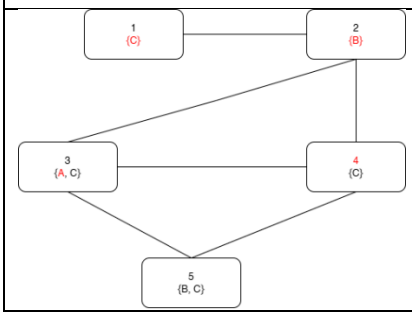
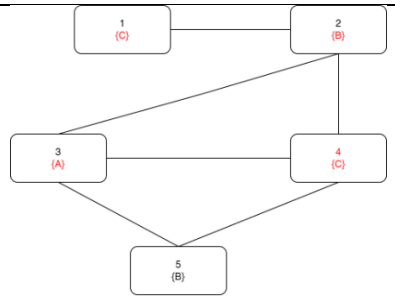
Now	Choose	Next Pos
		3

4) Choose A

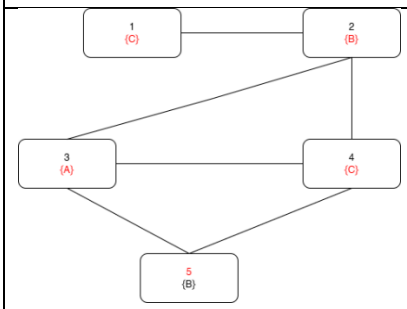
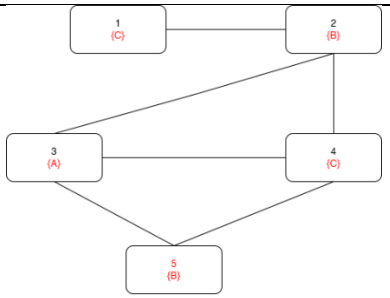
Now	Choose	Next Pos

		4

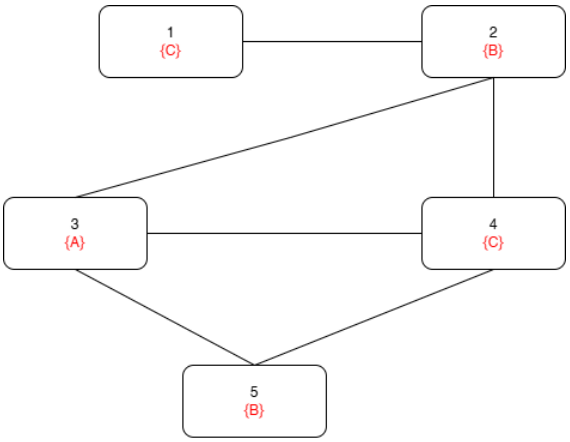
5) Choose C

Now	Choose	Next Pos
		5

6) Choose B

Now	Choose	Next Pos
		-

Jadi didapatkan untuk Class 1 = Professor C, Class 2 = Professor B, Class 3 = Professor A, Class 4 = Professor C, Class 5 = Professor B



3. Diketahui sebuah KB yang menggambarkan tentang pekerjaan, pelanggan, dan bos dengan symbol sebagai berikut:
   Occupation(p,o) : Person p has occupation o.
   Customer(p1,p2): Person p1 is a customer of person p2.
   Boss(p1,p2): Person p1 is a boss of person p2.
   Doctor, Surgeon, Lawyer, Actor, Emily, Joe merupakan konstanta.
   Ekspresikan kalimat di bawah ini dalam First-Order Logic:
   a. Emily is either a surgeon or a lawyer.
      Occupation(Emily, Surgeon) ∨ Occupation(Emily, Lawyer)
   b. Joe is an actor, but he also holds another jobs.
      Occupation(Joe, Actor) ∧ ∃x(Occupation(Joe, x) ∧ x ≠ Actor)
   c. Emily has a boss who is lawyer.
      ∃x (Boss(x, Emily) ∧ Occupation(x, Lawyer))
   d. There exist a lawyer all of whose customers are doctors.
      ∃x(Occupation(x, Lawyer) ∧ ∀y(Customer(y, x) ⟹ Occupation(y, Doctor)))

4. Everyone who is smart, studies, and attends class will be prepared. Everyone who is prepared will pass a test if it is fair. If a test isn't fair, no one will pass the test. At least one student passed the AI-exam. Every Informatics-ITS student is smart. Jono is an Informatics-ITS student and attends class.
   a. Buat FOL dari ilustrasi diatas.
      Everyone who is smart, studies, and attends class will be prepared.
      ∀x(Smart(x) ∧ Studies(x) ∧ AttendsClass(x) ⟹ Prepared(x))

      Everyone who is prepared will pass a test if it is fair.
      ∀x(Prepared(x) ⟹ ∃y(Test(y) ∧ Fair(y) ⟹ Pass(x, y)))

      If a test isn't fair, no one will pass the test.
      ∀x((Test(x) ∧ ¬Fair(x)) ⟹ ∀y¬Pass(y, x))

      At least one student passed the AI-exam
      ∃x(Student(x) ∧ Pass(x, AI-exam))
      Test(AI-exam)

      Every Informatics-ITS student is smart
      ∀x(Student(x) ∧ InformaticsITS(x) ⟹ Smart(x))

      Jono is an Informatics-ITS student and attends class
      Student(Jono) ∧ InformaticsITS(Jono) ∧ AttendsClass(Jono)

   b. Buat CNF dari FOL yang telah dibuat.
      I. ∀x(Smart(x) ∧ Studies(x) ∧ AttendsClass(x)) ⟹ Prepared(x)
         ∀x(**¬(**Smart(x) ∧ Studies(x) ∧ AttendsClass(x)) **∨** Prepared(x))
         ∀x((¬Smart(x) ∨ ¬Studies(x) ∨ ¬AttendsClass(x)) ∨ Prepared(x))
         ∀x(¬Smart(x) ∨ ¬Studies(x) ∨ ¬AttendsClass(x) ∨ Prepared(x))

¬Smart(x) ∨ ¬Studies(x) ∨ ¬AttendsClass(x) ∨ Prepared(x)

II. ∀x(Prepared(x) ⟹ ∃y(Test(y) ∧ Fair(y) ⟹ Pass(x, y)))
∀x(¬Prepared(x) ∨ ∃y(Test(y) ∧ Fair(y) ⟹ Pass(x, y)))
∀x(¬Prepared(x) ∨ ∃y(¬(Test(y) ∧ Fair(y)) ∨ Pass(x, y)))
∀x(¬Prepared(x) ∨ ∃y((¬Test(y) ∨ ¬Fair(y)) ∨ Pass(x, y)))
∀x(¬Prepared(x) ∨ (¬Test(f(x)) ∨ ¬Fair(f(x))) ∨ Pass(x, f(x))))
∀x(¬Prepared(x) ∨ ¬Test(f(x)) ∨ ¬Fair(f(x)) ∨ Pass(x, f(x)))
¬Prepared(x) ∨ ¬Test(f(x)) ∨ ¬Fair(f(x)) ∨ Pass(x, f(x))
¬Prepared(y) ∨ ¬Test(f(y)) ∨ ¬Fair(f(y)) ∨ Pass(y, f(y))

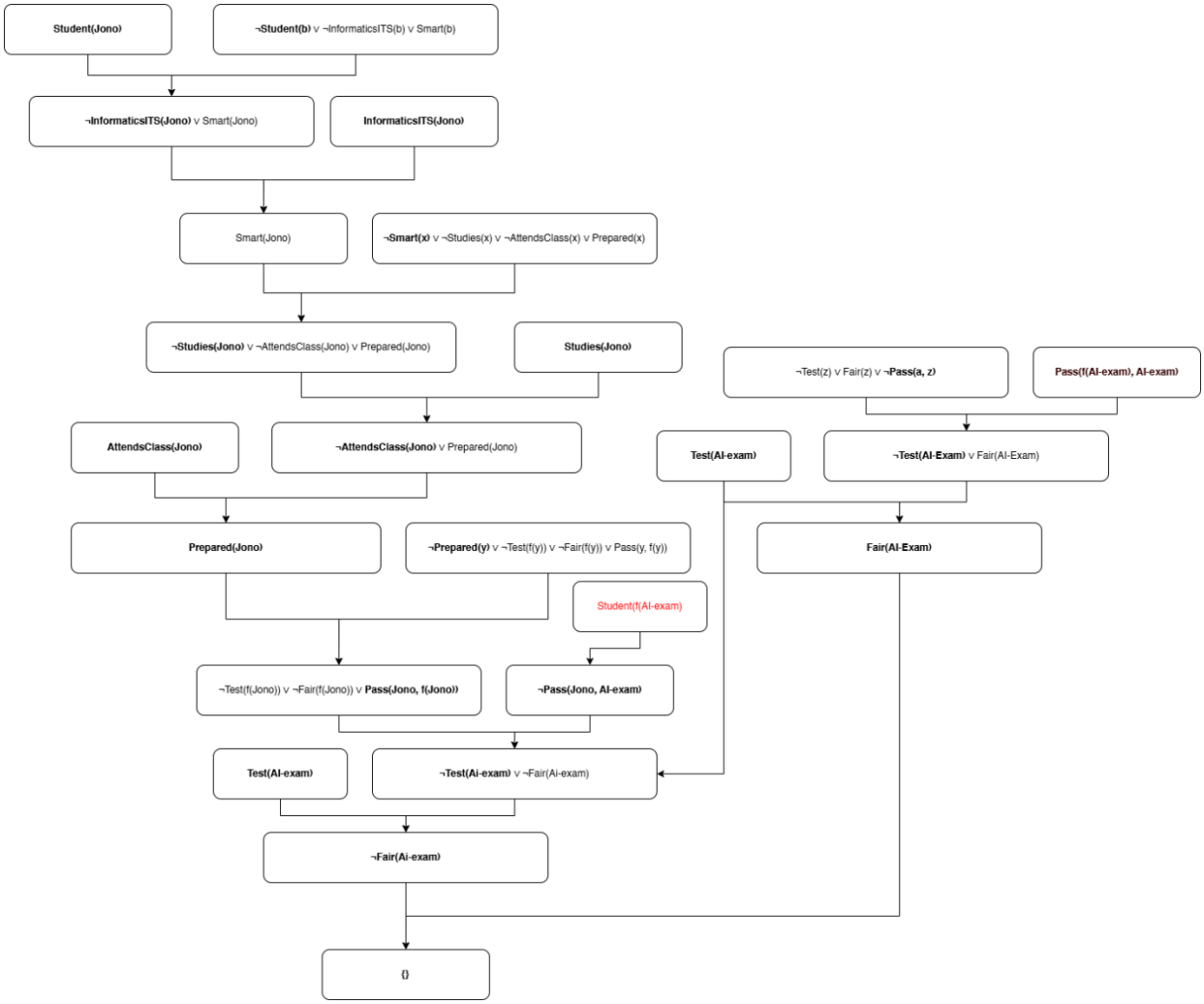
III. ∀x((Test(x) ∧ ¬Fair(x)) ⟹ ∀y¬Pass(y, x))
∀x(¬(Test(x) ∧ ¬Fair(x)) ∨ ∀y¬Pass(y, x))
∀x((¬Test(x) ∨ Fair(x)) ∨ ∀y¬Pass(y, x))
∀x((¬Test(x) ∨ Fair(x)) ∨ ¬Pass(y, x))
(¬Test(x) ∨ Fair(x)) ∨ ¬Pass(y, x))
¬Test(x) ∨ Fair(x) ∨ ¬Pass(y, x)
¬Test(z) ∨ Fair(z) ∨ ¬Pass(a, z)

IV. ∃x(Student(x) ∧ Pass(x, AI-exam))
Student(f(AI-exam)) ∧ Pass(f(AI-exam), AI-exam)
(1) Student(f(AI-exam))
(2) Pass(f(AI-exam), AI-exam)
(3) Test(AI-exam)

V. ∀x(Student(x) ∧ InformaticsITS(x) ⟹ Smart(x))
∀x(¬(Student(x) ∧ InformaticsITS(x)) ∨ Smart(x))
∀x((¬Student(x) ∨ ¬InformaticsITS(x)) ∨ Smart(x))
(¬Student(x) ∨ ¬InformaticsITS(x) ∨ Smart(x))
¬Student(b) ∨ ¬InformaticsITS(b) ∨ Smart(b)

VI. Student(Jono) ∧ InformaticsITS(Jono) ∧ AttendsClass(Jono)
(1) Student(Jono)
(2) InformaticsITS(Jono)
(3) AttendsClass(Jono)

c. Buktikan dengan resolusi bahwa: If Jono studies, he will pass AI-exam.
If Jono studies, he will pass AI-exam
Studies(Jono) ⟹ Pass(Jono, AI-exam)
¬Studies(Jono) ∨ Pass(Jono, AI-exam)
Start :
(1) Negasikan Hasil : ¬(¬Studies(Jono) ∨ Pass(Jono, AI-exam))
= Studies(Jono) ∧ ¬Pass(Jono, AI-exam)
= Studies(Jono)
= ¬Pass(Jono, AI-exam)
Buat Resolution Tree (Untuk Melihat Graph Lebih Baik Sebaiknya Gunakan Link Ini):
DrawIO
https://drive.google.com/file/d/1xQfcYxdXHLgQU0HTSOK1R5Qim3jkqOOS/view?usp=sharing
PNG
https://drive.google.com/file/d/1w46e6kmE-_LWjbM0SueRtM2cRAmVbIIV/view?usp=sharing


Student(Jono)
¬Student(b) ∨ ¬InformaticsITS(b) ∨ Smart(b)
¬InformaticsITS(Jono) ∨ Smart(Jono)
InformaticsITS(Jono)
Smart(Jono)
¬Smart(x) ∨ ¬Studies(x) ∨ ¬AttendsClass(x) ∨ Prepared(x)
¬Studies(Jono) ∨ ¬AttendsClass(Jono) ∨ Prepared(Jono)
Studies(Jono)
¬Test(z) ∨ Fair(z) ∨ ¬Pass(a, z)
Pass(f(AI-exam), AI-exam)
AttendsClass(Jono)
¬AttendsClass(Jono) ∨ Prepared(Jono)
Test(AI-exam)
¬Test(AI-Exam) ∨ Fair(AI-Exam)
Prepared(Jono)
¬Prepared(y) ∨ ¬Test(f(y)) ∨ ¬Fair(f(y)) ∨ Pass(y, f(y))
Fair(AI-Exam)
Student(f(AI-exam)
¬Test(f(Jono)) ∨ ¬Fair(f(Jono)) ∨ Pass(Jono, f(Jono))
¬Pass(Jono, AI-exam)
Test(AI-exam)
¬Test(AI-exam) ∨ ¬Fair(AI-exam)
¬Fair(AI-exam)
{}