1. Assignment

Exercise 1: [General] (6 Points)
Review the web or AI-literature to discover if computers can currently solve the following tasks and give a short explanation:
1. Detect humor in movies
2. Answer spoken questions about soccer
3. Find medical diagnoses
4. Find and prove mathematical theorems
5. Steal cycles from humans
6. Drive through town using a car, obeying traffic rules

Exercise 2: [Familiarize with RDF] (4 Points)
Familiarize with RDF and RDF Schema, by reading the specification of the W3C (http://www.w3.org/RDF). Have a look at the functionality of the RDF Validator (http://www.w3.org/RDF/Validator) (as a starting point you could visualize a few of the examples from the lecture as a graph).
Have a look at the underlying ontology of the Soccer News (http://z.about.com/6/g/worldsoccer/b/index.rdf).
Which resources do exist, which basic ontologies are used?

We wish you Happy Easter!

Submission deadline is Thursday, 28th April, at the chair before the lecture.
First tutorials will be held on 2nd May and 3rd May.
Methods of Artificial Intelligence, SS 2011

1. Assignment

Exercise 1: [General] (6 Points)
Review the web or AI-literature to discover if computers can currently solve the following tasks and give a short explanation:
1. Detect humor in movies
2. Answer spoken questions about soccer
3. Find medical diagnoses
4. Find and prove mathematical theorems
5. Steal cycles from humans
6. Drive through town using a car, obeying traffic rules

An interesting paper in this context is “Philosophy of Cognition – Why people think computers can’t” by Reut Tsarfaty (http://www.illc.uva.nl/~rtsarfat/MMAI.pdf).

1. Detect humor in movies: Possible to some extend. Computational Humor is a branch of computational linguistics and artificial intelligence and deals with humor generation as well as humor detection. There is a book about Computational Humor at Springer (http://www.springerlink.com/content/d1m5v3341gp7w674/)

2. Answer spoken questions about soccer: Possible. Berti the SoccerBot is a demonstrator at DFKI with whom visitors can interact via natural language. Berti answers questions about the latest sports results, match locations and the current national league team standings. Berti can be visited in the foyer of DFKI here in Saarbrücken.

3. Find medical diagnoses: Possible. MYCIN is such a system and was developed at Stanford University in the early 1970s. The system was designed to identify bacteria causing severe infections, such as bacteremia and meningitis, and to recommend antibiotics, with the dosage adjusted for patient’s body weight — the name derived from the antibiotics themselves, as many antibiotics have the suffix “-mycin” (Wikipedia).

4. Find and prove mathematical theorems: Possible. As an example, Automated Theorem Proving (ATP) was used to proof the Robbins Problem in 1996. The Robbins Problem was stated in 1933 by Herbert Robbins and neither he nor anyone else could proof it. An ATP system called EQP found the proof after 8 days of searching on an RS/600 processor. Prof. Harald Ganzinger (MPI Saarbrücken, died in 2004) and Prof. Jörg Siekmann (DFKI Saarbrücken) where/are leading scientists in the field of ATP.

5. Steal cycles from humans: The term „Stealing cycles from humans” is used in context with Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA). In a Turing Test, a human tries to indentify if he’s dealing with a computer or another human. A CAPTCHA can be seen as a reverted Turing Test, where a computer tries to identify if it’s
dealing with a human or another computer. CAPTCHAs are problems that are hard to solve for computers but easy to solve for humans (for example, distorted pictures of numbers and characters). A system that „steals (computational) cycles from humans“ captures these pictures, presents them to a human user to solve and promises something in return (access to a website or service for example).

6. Drive through town using a car, obeying traffic rules: Possible to some extent (2010). The third driverless car competition of the DARPA Grand Challenge was commonly known as the DARPA Urban Challenge. The course involved a 96 km urban area course, to be completed in less than 6 hours. Rules included obeying all traffic regulations while negotiating with other traffic and obstacles and merging into traffic. (Wikipedia)

- Driverless Car (http://en.wikipedia.org/wiki/Driverless_car#Fully_autonomous)
- Google Driverless Car (http://en.wikipedia.org/wiki/Google_driverless_car)

Articles in German:
- Sebastian Thrun: Der Fahrlehrer der Roboter-Autos
- Intelligentes Forschungsfahrzeug: Roboterauto "Leonie" rollt durch Braunschweig
- Google testet führerlose Fahrzeuge: Die Roboterautos sind los
  http://www.stern.de/digital/computer/google-testet-fuehrerlose-fahrzeuge-die-roboterautos-sind-los-1612412.html

Exercise 2: [Familiarize with RDF] (4 Points)
Familiarize with RDF and RDF Schema, by reading the specification of the W3C (http://www.w3c.org/RDF). Have a look at the functionality of the RDF Validator (http://www.w3c.org/RDF/Validator) (as a starting point you could visualize a few of the examples from the lecture as a graph).

Have a look at the underlying ontology of the Soccer News (http://z.about.com/6/g/worldsoccer/b/index.rdf). Which resources do exist, which basic ontologies are used?

Existing resources:

Used Ontologies:
<rdf:rdf
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:sy="http://purl.org/rss/1.0/modules/syndication/"
 xmlns:admin="http://webns.net/mvcb/"
 xmlns:cc="http://web.resource.org/cc/"
 xmlns ="http://purl.org/rss/1.0/">
Methods of Artificial Intelligence, SS 2011

2. Assignment

Exercise 1: [Model a Soccer-Team Ontology in RDF] (6 Points)
For this exercise you have to model the local football club 1. FC Saarbrücken by drawing a diagram. Your task is to (partially) annotate the semantic information about the team using RDF, to represent the statements below. You should make use of special resources (like bags, sequences or alternatives). It is a good idea to start with an informal diagram. If you implement this ontology in the RDF syntax, you can generate the graph automatically with the RDF Validator (see 1. Assignment).

- The club is based in Saarbrücken, Saarland.
- Manager: Paul Borgard is the President of the Club Committee, and Reinhard Klimmt is the Chairman of the Advisory Board.
- Jürgen Luginger is the coach of the team.
- The club played in the Second French Division, the First German League and the Fifth German League.
- Ufuk Özbek, Giuseppe Pisano, and Alexander Otto are team players.
- Otto Rehhagel, Klaus Schlappner and Tom Dooley are former famous coaches.
- Manager, coach and players are members of the 1. FC Saarbrücken.
- Every player is trained by the coach.

Exercise 2: [Reification in RDF] (4 Points)
Draw the diagrams representing the following statements in an RDF data model:

2. “Bill believes that the creator of the resource http://www.google.com is Peter Norvig.”

Hint: Use the namespace xmlns:s="http://description.org/schema/" for the predicates Creator and believedBy.

Exercise 3: [T-Box and A-box] (2 Points)
Describe in your own words what kind of knowledge is represented in the T-Box and in the A-Box of a terminological system.
Exercise 4: [Taxonomy] (8 Points)
Build a taxonomy in graphical representation (in analogy to the lecture) for different kinds of food, that contains primitive and disjunct classes, partitions and overlappings. Mark them as in the lecture.

Exercise 7: [Roles] (6 Points)
Draw a graphical representation of the following roles:
Public services are financed through taxes that are paid by citizens, who use the public services.

Submission deadline is Thursday, 5th May, at the chair before the lecture.

The Thursday lecture, 5th May 2011, will be held 14:15 - 15:45
at DFKI, Building D3_2, Room -2.17 "Reuse".
Methods of Artificial Intelligence, SS 2011

2. Assignment

Exercise 1: [Model a Soccer-Team Ontology in RDF] (6 Points)
For this exercise you have to model the local football club *1. FC Saarbrücken* by drawing a diagram. Your task is to (partially) annotate the semantic information about the team using RDF, to represent the statements below. You should make use of special resources (like bags, sequences or alternatives). It is a good idea to start with an informal diagram. If you implement this ontology in the RDF syntax, you can generate the graph automatically with the RDF Validator (see 1. Assignment).

- The club is based in Saarbrücken, Saarland.
- Manager: Paul Borgard is the President of the Club Committee, and Reinhard Klimmt is the Chairman of the Advisory Board.
- Jürgen Luginger is the coach of the team.
- The club played in the Second French Division, the First German League and the Fifth German League.
- Ufuk Özbek, Giuseppe Pisano, and Alexander Otto are team players.
- Otto Rehhagel, Klaus Schlappner and Tom Dooley are former famous coaches.
- Manager, coach and players are members of the *1. FC Saarbrücken*.
- Every player is trained by the coach.

```xml
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://www.fc-saarbruecken.de#">
  <rdf:Description rdf:about="http://www.PaulBorgard.de">
    <dc:position>The President of the Club Committee</dc:position>
    <dc:sbrelated>Manager of the Saarbruecken Football Club</dc:sbrelated>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.ReinhardKlimmt.de">
    <dc:position>Chairman of the Advisory Board</dc:position>
    <dc:sbrelated>Manager of the Saarbruecken Football Club</dc:sbrelated>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.fc-saarbruecken.de/teamplayers">
    <dc:trainedby>Coach</dc:trainedby>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.fc-saarbruecken.de/coach">
    <dc:trains>Team Players</dc:trains>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.fc-saarbruecken.de">
    <dc:Leagues>
      <rdf:Alt>
        <rdf:li>Second French Division</rdf:li>
        <rdf:li>First German League</rdf:li>
        <rdf:li>Fifth German League</rdf:li>
      </rdf:Alt>
  </dc:Leagues>
</rdf:RDF>
```
<dc:Manager>
  <rdf:Seq>
    <rdf:li>Paul Borgard</rdf:li>
    <rdf:li>Reinhard Klimmt</rdf:li>
  </rdf:Seq>
</dc:Manager>

<dc:Coach> Jürgen Luginger </dc:Coach>

<dc:TeamPlayers>
  <rdf:Bag>
    <rdf:li>Ufuk Özbek</rdf:li>
    <rdf:li>Giuseppe Pisano</rdf:li>
    <rdf:li>Alexander Otto</rdf:li>
  </rdf:Bag>
</dc:TeamPlayers>

<dc:FormerCoaches>
  <rdf:Bag>
    <rdf:li>Otto Rehagel</rdf:li>
    <rdf:li>Klaus Schlappner</rdf:li>
    <rdf:li>Tom Dooley</rdf:li>
  </rdf:Bag>
</dc:FormerCoaches>

<dc:Headquaters> Saarbruecken, Saarland </dc:Headquaters>

<dc:MembersoftheFootballClub>
  <rdf:Bag>
    <rdf:li>Team Players</rdf:li>
    <rdf:li>Trainer</rdf:li>
    <rdf:li>Manager</rdf:li>
  </rdf:Bag>
</dc:MembersoftheFootballClub>
</rdf:Description>
</rdf:RDF>
Exercise 2: [Reification in RDF] (4 Points)
Draw the diagrams representing the following statements in an RDF data model:

2. “Bill believes that the creator of the resource http://www.google.com is Peter Norvig.”

**Hint:** Use the namespace `xmlns:s="http://description.org/schema/"` for the predicates `Creator` and `believedBy`.

1. The creator of the resource http://www.google.com is Larry Page

```xml
<?xml version="1.0"?><rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:s="http://description.org/schema/">
  <rdf:Description rdf:about="http://www.google.com">
    <s:Creator>Peter Norvig</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

2. Bill believes that the creator of the resource http://www.google.com is Peter Norvig

```xml
<?xml version="1.0"?><rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:s="http://description.org/schema/">
  <rdf:Description rdf:type="rdf:Statement">
    <s:believedBy>Bill</s:believedBy>
    <r rdf:predicate="s:Creator">
      <r rdf:subject="http://www.google.com">
        Peter Norvig</r>
    </r>
  </rdf:Description>
</rdf:RDF>
```

Exercise 3: [T-Box and A-box] (2 Points)
Describe in your own words what kind of knowledge is represented in the T-Box and in the A-Box of a terminological system.

Example:
The T-Box contains the vocabulary; the A-Box contains statements about that vocabulary.
Or
In case of object oriented programming languages: The T-Box contains classes and the A-Box contains instances of these classes.

Exercise 4: [Taxonomy] (8 Points)
Build a taxonomy in graphical representation (in analogy to the lecture) for different kinds of food, that contains primitive and disjunct classes, partitions and overlappings. Mark them as in the lecture.

Sample Solution:
Exercise 5: [Roles] (6 Points)
Draw a graphical representation of the following roles:
Public services are financed through taxes that are paid by citizens, who use the public services.
Exercise 0: [OWL] (0 Points)

Exercise 1: [OWL] (5 Points)
Given is the following ontology:

- **Elephant** is a sub-concept of the concept **Animal**.
- **Herbivore** is a concept whose members are exactly those animals who eat only plants.
- **Adult_Elephant** is a concept whose members are exactly those elephants older than 20 years.
- Adult elephants might have children who are again elephants.
- **Djan** is an elephant.
- **Djan** is 23 years old.
- **Djan** is the child of **Zorba**.

1. Represent the ontology in OWL.
2. Which of these statements belong to the T-Box and which to the A-Box? Provide another A-Box modeling another elephant using the same T-Box.

Exercise 2: [Bipolar Inheritance Networks] (5 Points)
Given is the following bipolar inheritance network:

```
      educated
     /       \
academic----sportsman
       \       /
         PhD----professional Boxer
               \   /
                Vitali
```

a) Determine the credulous extensions.
b) Determine the skeptical extension.
Exercise 3: [Bipolar Inheritance Networks] (10 Points)

Determine for the following inheritance networks, whether they have conflicts and how they can be solved in credulous extensions (Contradiction, Off-path pre-emption, On-path pre-emption). Explain your solution briefly.

Submission deadline is Thursday, 12th May, at the chair before the lecture.

The Wednesday lecture, 18th May 2011, is cancelled and scheduled Tuesday, 17th May 2011, 16:15 – 17:45 at Building E1_3, HS 001.
Exercise 1: [OWL] (5 Points)
Given is the following ontology:

- Elephant is a sub-concept of the concept Animal.
- Herbivore is a concept whose members are exactly those animals who eat only plants.
- Adult_Elephant is a concept whose members are exactly those elephants older than 20 years.
- Adult elephants might have children who are again elephants.
- Djan is an elephant.
- Djan is 23 years old.
- Djan is the child of Zorba.

1. Represent the ontology in OWL.
2. Which of these statements belong to the T-Box and which to the A-Box? Provide another A-Box modeling another elephant using the same T-Box.

1. Ontology in OWL

```owl
<?xml version="1.0"?>
<owl:Ontology rdf:about=""/> 
<owl:Class rdf:ID="Animal"/>
<owl:Class rdf:ID="Herbivore">
<owl:equivalentClass>
<owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:about="#Animal"/>
<owl:Restriction>
<owl:onProperty>
<owl:SymmetricProperty rdf:ID="eats"/>
</owl:onProperty>
<owl:allValuesFrom>
<owl:Class rdf:ID="Plants"/>
</owl:allValuesFrom>
</owl:Restriction>
</owl:intersectionOf>
</owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="Adult_Elephant">
<owl:equivalentClass>
<owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:ID="Elephant"/>
<owl:Restriction>
</owl:intersectionOf>
</owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="Djan">
<owl:equivalentClass>
<owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:ID="Elephant"/>
<owl:Restriction>
</owl:intersectionOf>
</owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="Zorba">
<owl:equivalentClass>
<owl:intersectionOf rdf:parseType="Collection">
<owl:Class rdf:ID="Elephant"/>
<owl:Restriction>
</owl:intersectionOf>
</owl:equivalentClass>
</owl:Class>
```

<owl:Class rdf:about="#Elephant">
  <rdfs:subClassOf rdf:resource="#Animal"/>
</owl:Class>

<owl:ObjectProperty rdf:ID="hasChild">
  <rdfs:range rdf:resource="#Elephant"/>
  <rdfs:domain rdf:resource="#Adult_Elephant"/>
  <owl:inverseOf>
    <owl:ObjectProperty rdf:ID="hasParent"/>
  </owl:inverseOf>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#hasParent">
  <rdfs:range rdf:resource="#Adult_Elephant"/>
  <rdfs:domain rdf:resource="#Elephant"/>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:about="#age">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#eats">
  <owl:inverseOf rdf:resource="#eats"/>
  <rdfs:domain rdf:resource="#Animal"/>
</owl:DatatypeProperty>

<owl:DataRange>
  <owl:oneOf rdf:parseType="Resource">
    <rdf:rest rdf:parseType="Resource">
      <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string">meat</rdf:first>
    </rdf:rest>
  </owl:oneOf>
  <owl:oneOf rdf:parseType="Resource">
    <rdf:rest rdf:parseType="Resource">
      <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string">meat and plants</rdf:first>
    </rdf:rest>
  </owl:oneOf>
  <owl:oneOf rdf:parseType="Resource">
    <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
  </owl:oneOf>
</owl:DataRange>

<Plants rdf:ID="Tomato"/>
<Elephant rdf:ID="Djan">
  <age rdf:datatype="http://www.w3.org/2001/XMLSchema#int">23</age>
  <hasParent>
    <Adult_Elephant rdf:ID="Zorba">
      <hasChild rdf:resource="#Djan"/>
    </Adult_Elephant>
  </hasParent>
</Elephant>
<Plants rdf:ID="Eggplant"/>
</rdf:RDF>
2. Which of these statements belong to the T-Box and which to the A-Box? 
   Provide another A-Box modeling another elephant using the same T-Box.

   The first four statements belong to the T-Box and the rest to the A-Box.

   Another A-Box would be:
   - Jumbo is an elephant
   - Mambo is the child of Jumbo
   - Jumbo is 29 years old

Exercise 2: [Bipolar Inheritance Networks] (5 Points)
Given is the following bipolar inheritance network:

a) Determine the credulous extensions.

   \[
   \Phi_1 = \Gamma \cup \{ \text{P}\rightarrow \text{A}\rightarrow \text{e}; \text{B}\rightarrow \text{S}\rightarrow \text{e}; \text{V}\rightarrow \text{P}\rightarrow \text{A}; \text{V}\rightarrow \text{B}\rightarrow \text{S}; \text{V}\rightarrow \text{P}\rightarrow \text{A}\rightarrow \text{e} \}
   \]

   \[
   \Phi_2 = \Gamma \cup \{ \text{P}\rightarrow \text{A}\rightarrow \text{e}; \text{B}\rightarrow \text{S}\rightarrow \text{e}; \text{V}\rightarrow \text{P}\rightarrow \text{A}; \text{V}\rightarrow \text{B}\rightarrow \text{S}; \text{V}\rightarrow \text{B}\rightarrow \text{S}\rightarrow \text{e} \}
   \]

   \[
   \Phi_3 = \Gamma \cup \{ \text{P}\rightarrow \text{A}\rightarrow \text{e}; \text{B}\rightarrow \text{S}\rightarrow \text{e}; \text{V}\rightarrow \text{B}\rightarrow \text{A}; \text{V}\rightarrow \text{B}\rightarrow \text{S}; \text{V}\rightarrow \text{B}\rightarrow \text{S}\rightarrow \text{e} \}
   \]

b) Determine the skeptical extension.

   \[
   \Phi_S = \Gamma \cup \{ \text{P}\rightarrow \text{A}\rightarrow \text{e}; \text{B}\rightarrow \text{S}\rightarrow \text{e}; \text{V}\rightarrow \text{B}\rightarrow \text{S}; \text{V}\rightarrow \text{B}\rightarrow \text{S}\rightarrow \text{e} \}
   \]

   \text{(Remark: use shortcuts for the names of the nodes: V, P, B(\neg\text{Boxer}), A, S, e)}
Exercise 3: [Bipolar Inheritance Networks] (10 Points)

Determine for the following inheritance networks, whether they have conflicts and how they can be solved in credulous extensions (Contradiction, Off-path pre-emption, On-path pre-emption). Explain your solution briefly.

(1) On-Path-Pre-Emption:
Pre-empted path: A → B → C → E → D, A → B → E → D
Pre-Emptor: A → B → C → D
Redundant: B → E

(2) Contradiction
Contradicting Paths: C → E → B → A and C → E → D → A

(3) On-Path-Pre-Emption:
Pre-Empted Path: E → B → C → D → A
Pre-Emptor: E → B → A

(4) none

(5) On and Off-Path-Pre-Emption:
Pre-Empted Paths: D → A → C → B (off), D → E → C → B (on)
Pre-Emptor: D → E → B
Exercise 1: [Conflict Resolution with meta rules] (6 Points)
Peter wants to watch a funny movie for adults, i.e. the movie must not be boring or sad.

Working Memory:
- W1: “Summer of Tigers” is a movie about tigers and it’s not a cartoon.
- W2: “Happy Feet” is a cartoon about tigers.
- W3: “Barka the Tiger” is a movie about tigers and not a cartoon.
- W4: Anna watched “Barka the Tiger”.
- W5: R3 is entered by Anna.
- W6: R4 is entered by Mike.
- W7: R5 is entered by Elena.
- W8: Anna and Peter are friends.
- W9: Mike knows a lot about movies.
- W10: Elena is a movie expert.
- W11: Anna has no idea about movies.

R1: If a movie is funny, not boring, not sad and for adults, then Peter will watch it.
   STOP
R2: If a movie is a cartoon, then it’s for children, otherwise for adults.
R3: If a movie is about tigers, then it’s funny.
R4: If a movie is about tigers, then it’s boring.
R5: If a movie is about tigers, then it’s sad.

MR1: Prefer rules entered by experts.
MR2: Prefer rules entered by friends, if they know the movie.

MMR1: Friends know your taste for movies better than any expert.

Determine with these rules and facts which movie Peter is going to watch. Write down the rules and to which working memory elements they apply, the resulting conflict sets and how they are resolved. (Forward Chaining)
Exercise 2: [Formal Conflict Resolution] (8 Points)
Have a look at the production system below with the following working memory WM:

WM ( (P S) (Q T) (P T) (R Z) (Q S) . . . (W V) (W T) )

The elements of WM are ordered from left to right with respect to their increasing dwell time. Elements (P S) and (Q T) where added in last computing cycle, (P T), (R Z) and (Q S) in the penultimate cycle (two cycles ago) and (W V) and (W T) 100 cycles ago.

The production rules are in order of their storage time:

P1 ( (P =X) (=Y S) (W =X) ) → ...
P2 ( (=X S) (=X T) ¬(K L) ) → ...
P3 ( (=X S) (=X T) (R Z) (W =Y) ¬(K L) ) → ...
P4 ( (Q =X) (P =X) ¬(M N) (W V) ¬(N Z) ) → ...

1. Write down all instantiations of each rule with the given WM.
2. For each specialization relation SC from the lecture, write down which instantiations are preferred. Justify your answer.

Exercise 3: [Production Systems] (8 Points)
The following production system is designed to calculate the first 7 Fibonacci-Numbers. Here, PLUS(X, Y) calculates the sum of X and Y, SUCC(X) calculates the successor of X (in other words: X+1) and PRINT(X) prints out the value of X on the console:

WM: (START) (P 0) (Q 1) (END 5)

R1: (START) & (P X1) & (Q X2) → DELETE (START), ADD (COUNT 0), ADD (TEMP 0), ADD (CALC), PRINT (X1 X2)
R2: (END X1) & (COUNT X1) → STOP
R3: (CALC) & (P X1) & (Q X2) & (TEMP X3) → DELETE (CALC), REPLACE (X3, PLUS (X1, X2), (TEMP X3)), ADD (PRINTOUT)
R4: (PRINTOUT) & (TEMP X3) & (COUNT X4) → DELETE (PRINTOUT), ADD (SHIFT), PRINT (X3), REPLACE (X4, SUCC(X4),(COUNT X4))
R5: (SHIFT) & (P X1) & (Q X2) & (TEMP X3) → DELETE (SHIFT), REPLACE (X1, X2, (P X1)), REPLACE (X2, X3, (Q X2))

1. Calculate the output of this production system until no more rules are left to fire. For each cycle, write down the contents of the WM, the conflict set, the instantiation of the firing rule, the output (if there is any) and the resulting WM.

Example:
WM: (START) (P 0) (Q 1) (END 5)
Conflict set: {R1}
Instantiation: [R1 ((START) (P 0) (Q 1))]
Output: 0 1
WM: (COUNT 0) (TEMP 0) (CALC) (P 0) (Q 1) (END 5)
Conflict set: ...

2. Obviously there is an error in the rules above: Rule R2 is never going to fire. Change rule R5 in such a way that the cycle R3, R4, R5 is repeated until R2 fires.
3. The rules above are independent of the chosen conflict resolution strategy. Why?
Exercise 4: [RETE] (18 Points)
Tom, Peter, Anna and Maria are watching horse races and make bets on different horses.

1. Use the given RETE network and the entries of the table below to identify the winner.
   Apply the RETE algorithm as described in the lecture and fill the alpha and beta memory
   nodes in the graph below.

2. Answer these questions:
   • What sense does the timestamp restriction make in this horse bidding scenario?
   • How do you interpret the meaning of the first 3 nodes applied to the class type “Bet”?
   • Where in the RETE network can you see which bets correctly predicted the winners of
     the first 2 races?

3. Write down the conditions of the rule represented by this RETE network.

4. Write down the conditions of 3 new rules, each of which check whether a bet correctly
   predicted the winner of one of the 3 races. (Remark: You don’t have to use the timestamp
   restriction and the conditions that check whether the tips are in a certain subset of horses)

5. Draw RETE networks for these rules and merge them into one.

Table of bets: (e.g. row 1: Tom makes a bet that horse 2 wins the first race, horse 6 wins the
second race and horse number 1 wins the third race)

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Class</th>
<th>Name</th>
<th>Tip1</th>
<th>Tip2</th>
<th>Tip3</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Bet</td>
<td>Tom</td>
<td>Horse2</td>
<td>Horse6</td>
<td>Horse1</td>
</tr>
<tr>
<td>17</td>
<td>Bet</td>
<td>Peter</td>
<td>Horse2</td>
<td>Horse9</td>
<td>Horse8</td>
</tr>
<tr>
<td>43</td>
<td>Bet</td>
<td>Anna</td>
<td>Horse2</td>
<td>Horse4</td>
<td>Horse1</td>
</tr>
<tr>
<td>51</td>
<td>Bet</td>
<td>Maria</td>
<td>Horse2</td>
<td>Horse4</td>
<td>Horse2</td>
</tr>
</tbody>
</table>

Table of races. (e.g. row 1: Horse number 9 won the pre-race and horse number 10 was second)

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Class</th>
<th>Type</th>
<th>Winner1</th>
<th>Winner2</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>Race</td>
<td>Pre-Race</td>
<td>Horse9</td>
<td>Horse10</td>
</tr>
<tr>
<td>65</td>
<td>Race</td>
<td>Race1</td>
<td>Horse2</td>
<td>Horse1</td>
</tr>
<tr>
<td>80</td>
<td>Race</td>
<td>Race2</td>
<td>Horse6</td>
<td>Horse8</td>
</tr>
<tr>
<td>99</td>
<td>Race</td>
<td>Race3</td>
<td>Horse1</td>
<td>Horse8</td>
</tr>
</tbody>
</table>
Submission deadline is this Thursday (19th of May) before the lecture.
Exercise 1: [Conflict Resolution with meta rules] (6 Points)
Peter wants to watch a funny movie for adults, i.e. the movie must not be boring or sad.

Working Memory:
W1: “Summer of Tigers” is a movie about tigers and it’s not a cartoon.
W2: “Happy Feet” is a cartoon about tigers.
W3: “Barka the Tiger” is a movie about tigers and not a cartoon.
W4: Anna watched “Barka the Tiger”.
W5: R3 is entered by Anna.
W6: R4 is entered by Mike.
W7: R5 is entered by Elena.
W8: Anna and Peter are friends.
W9: Mike knows a lot about movies.
W10: Elena is a movie expert.
W11: Anna has no idea about movies.

R1: If a movie is funny, not boring, not sad and for adults, then Peter will watch it.
STOP
R2: If a movie is a cartoon, then it’s for children, otherwise for adults.
R3: If a movie is about tigers, then it’s funny.
R4: If a movie is about tigers, then it’s boring.
R5: If a movie is about tigers, then it’s sad.

MR1: Prefer rules entered by experts.
MR2: Prefer rules entered by friends, if they know the movie.

MMR1: Friends know your taste for movies better than any expert.

Determine with these rules and facts which movie Peter is going to watch. Write down the rules and to which working memory elements they apply, the resulting conflict sets and how they are resolved. (Forward Chaining)

This is the intuitive, informal solution, which we accepted:

R2 applied to W1 → W21: „Summer of Tigers“ is a movie for adults.
R2 applied to W2 → W22: „Happy Feet“ is a movie for children.
R2 applied to W3 → W23: „Barka the Tiger“ is a movie for adults.

R3 applied to W1 → W12: „Summer of Tigers“ is a funny movie.
R3 applied to W2 → W13: „Happy Feet“ is a funny movie.
R3 applied to W3 → W14: „Barka the Tiger“ is a funny movie.
R4 applied to W1 → W15: „Summer of Tigers“ is a boring movie.
R4 applied to W2 → W16: „Happy Feet“ is a boring movie.
R4 applied to W3 → W17: „Barka the Tiger“ is a boring movie.

R5 applied to W1 → W18: „Summer of Tigers“ is a sad movie.
R5 applied to W2 → W19: „Happy Feet“ is a sad movie.
R5 applied to W2 → W20: „Barka the Tiger“ is a sad movie.

Conflict set 1: {R3, R4, R5}
Conflict resolution 1: MR1 → R5, MR2 → R3

Conflict set 2: {R3, R5}
Conflict resolution 2: MMR1 → MR2 → R3 → {„Barka the Tiger“}

Alternative Solution:
Using this amendment:
The production system applies rules to the working memory as follows:

WM := initial knowledge
UNTIL P is solved or no production is left to fire DO:
(1) Determine set K of all rules whose premises are fulfilled by WM
(2) Determine conflict set C, which contains all rules in K with the same premise/condition
(3) Select rule R out of C according to meta rules and discard the remaining conflicting rules from K
(4) Apply all possible instantiations of rules in K in their given order (R1 before R2, R2 before R3 and so on)
(Meta-Rules: Meta-Rules are in conflict if more than one meta-rule can be applied to a conflicting set C of rules with different results.)

Solution:
K = {R2, R3, R4, R5}, C = {R3, R4, R5} → selected Rule: R3 (MR2 due to WM5 and WM8)
or R5 (MR1 due to WM7 and WM10)
MMR applied (due to W4 and W8): R3 -> K = {R2, R3}
R2 applied to W1 → W21: „Summer of Tigers“ is a movie for adults.
R2 applied to W2 → W22: „Happy Feet“ is a movie for children.
R2 applied to W3 → W23: „Barka the Tiger“ is a movie for adults.

(R3 applied to W1 → W12: „Summer of Tigers“ is a funny movie.
R3 applied to W2 → W13: „Happy Feet“ is a funny movie.)
R3 applied to W3 → W14: „Barka the Tiger“ is a funny movie.

K = {R1....} → Peter watches “Barka the Tiger” (because of W23 and W14)
→ Peter watches “Summer of Tiger” (because of W21 and W12)

The rules and instantiations in italic are part of the result if R3 is applied to all movies. MR2 would actually only apply to R3 for the instantiation with “Barka the Tiger”, since Anna did not watch the other movies.
Exercise 2: [Formal Conflict Resolution] (8 Points)
Have a look at the production system below with the following working memory WM:

WM ( (P S) (Q T) (P T) (R Z) (Q S) . . . (W V) (W T) )

The elements of WM are ordered from left to right with respect to their increasing dwell time. Elements (P S) and (Q T) where added in last computing cycle, (P T), (R Z) and (Q S) in the penultimate cycle (two cycles ago) and (W V) and (W T) 100 cycles ago.

The production rules are in order of their storage time:

P1 ((P =X) (=Y S) (W =X) ) → … )
P2 ((=X S) (=X T) ¬(K L) ) → … )
P3 ((=X S) (=X T) (R Z) (W =Y) ¬(K L) ) → … )
P4 ((Q =X) (P =X) ¬(M N) (W V) ¬(N Z) ) → … )

1. Write down all instantiations of each rule with the given WM.

I11 X:=T, Y:=Q (P T) (Q S) (W T)
I12 X:=T, Y:=P (P T) (P S) (W T)
I21 X:=P (P S) (P T)
I22 X:=Q (Q S) (Q T)
I31 X:=P, Y:=V (P S) (P T) (R Z) (W V)
I32 X:=Q, Y:=V (Q S) (Q T) (R Z) (W V)
I33 X:=P, Y:=T (P S) (P T) (R Z) (W T)
I34 X:=Q, Y:=T (Q S) (Q T) (R Z) (W T)
I41 X:=T (Q T) (P T) (W V)
I42 X:=S (Q S) (P S) (W V)

2. For each specialization relation SC from the lecture, write down which instantiations are preferred. Justify your answer.

SC1:

P3 is a specialization of P2, because
- P3 has more premises than P2
- For every premise in P2, that contains constants, there is a respective premise in P3, that contains these constants as a subset (the first two premises)
- P3 and P2 are not identical.

There are no specializations in the sense of SC1 between the other productions

SC2:

- Here, an instantiation must have the data elements of another instantiation as a true subset to be a specialization.
- I12, I31 und I33 are specializations of I21
- I33 is a specialization of I11
- I32 and I34 are specializations of I22

SC3:

According to SC3, I12 is no specialization of I21, since production P1 does not have more premises as P2. The rest is the same as with SC2.
- I31 and I33 are specializations of I21
- I33 is specialization of I12
- I32 and I34 are specializations of I22
Exercise 3: [Production Systems] (8 Points)
The following production system is designed to calculate the first 7 Fibonacci-Numbers. Here, PLUS(X, Y) calculates the sum of X and Y, SUCC(X) calculates the successor of X (in other words: X+1) and PRINT(X) prints out the value of X on the console:

```
WM: (START) (P 0) (Q 1) (END 5)
R1: (START) & (P X1) & (Q X2) →
    DELETE (START), ADD (COUNT 0), ADD (TEMP 0), ADD (CALC),
    PRINT (X1 X2)
R2: (END X1) & (COUNT X1) → STOP
R3: (CALC) & (P X1) & (Q X2) & (TEMP X3) →
    DELETE (CALC), REPLACE (X3, PLUS (X1, X2), (TEMP X3)),
    ADD (PRINTOUT)
R4: (PRINTOUT) & (TEMP X3) & (COUNT X4) →
    DELETE (PRINTOUT), ADD (SHIFT), PRINT (X3),
    REPLACE (X4, SUCC(X4),(COUNT X4))
R5: (SHIFT) & (P X1) & (Q X2) & (TEMP X3) →
    DELETE (SHIFT), REPLACE (X1, X2, (P X1)),
    REPLACE (X2, X3, (Q X2))
```

1. Calculate the output of this production system until no more rules are left to fire. For each cycle, write down the contents of the WM, the conflict set, the instantiation of the firing rule, the output (if there is any) and the resulting WM.

   Example:
   WM: (START) (P 0) (Q 1) (END 5)
   Conflict set: {R1}
   Instantiation: [R1 ((START) (P 0) (Q 1))]
   Output: 0 1

   Step1:
   WM: (START) (P 0) (Q 1) (END 5)
   Conflict Set: {R1}
   Instantiation: [R1 ((START) (P 0) (Q 1))]
   Output: 0 1

   Step2:
   WM: (CALC) (TEMP 0) (COUNT 0) (P 0) (Q 1) (END 5)
   Conflict Set: {R3}
   Instantiation: [R3 ((CALC) (P 0) (Q 1) (TEMP 0))]
   Output:

   Step3:
   WM: (PRINTOUT) (TEMP 1) (COUNT 0) (P 0) (Q 1) (END 5)
   Conflict Set: {R4}
   Instantiation: [R4 ((PRINTOUT) (TEMP 1) (COUNT 0))]
   Output: 1

2. Obviously there is an error in the rules above: Rule R2 is never going to fire. Change rule R5 in such a way that the cycle R3, R4, R5 is repeated until R2 fires.

3. The rules above are independent of the chosen conflict resolution strategy. Why?
2. The Rule has to be changed in such a way that after shifting the numbers a new calculation step begins:

R5:  (SHIFT) & (P X1) & (Q X2) & (TEMP X3) →
    DELETE (SHIFT), REPLACE (X1, X2, (P X1)), REPLACE (X2, X3, (Q X2)),
    ADD (CALC)

3. The rules are independent of the chosen conflict resolution strategy because every rule (or step in the calculation) has a special “label” (e.g. CALC, START, etc.). These labels are always deleted after a rule has fired and a new label is added to trigger firing the next rule in the calculation process.
Exercise 4: [RETE] (18 Points)
Tom, Peter, Anna and Maria are watching horse races and make bets on different horses.

1. Use the given RETE network and the entries of the table below to identify the winner. 
   Apply the RETE algorithm as described in the lecture and fill the alpha and beta memory nodes in the graph below.

2. Answer these questions:
   • What sense does the timestamp restriction make in this horse bidding scenario?
   • How do you interpret the meaning of the first 3 nodes applied to the class type “Bet”?
   • Where in the RETE network can you see which bets correctly predicted the winners of the first 2 races?

3. Write down the conditions of the rule represented by this RETE network.

4. Write down the conditions of 3 new rules, each of which check whether a bet correctly predicted the winner of one of the 3 races. (Remark: You don’t have to use the timestamp restriction and the conditions that check whether the tips are in a certain subset of horses)

5. Draw RETE networks for these rules and merge them into one.

Table of bets: (e.g. row 1: Tom makes a bet that horse 2 wins the first race, horse 6 wins the second race and horse number 1 wins the third race)

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Class</th>
<th>Name</th>
<th>Tip1</th>
<th>Tip2</th>
<th>Tip3</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Bet</td>
<td>Tom</td>
<td>Horse2</td>
<td>Horse6</td>
<td>Horse1</td>
</tr>
<tr>
<td>17</td>
<td>Bet</td>
<td>Peter</td>
<td>Horse2</td>
<td>Horse9</td>
<td>Horse8</td>
</tr>
<tr>
<td>43</td>
<td>Bet</td>
<td>Anna</td>
<td>Horse2</td>
<td>Horse4</td>
<td>Horse1</td>
</tr>
<tr>
<td>51</td>
<td>Bet</td>
<td>Maria</td>
<td>Horse2</td>
<td>Horse4</td>
<td>Horse2</td>
</tr>
</tbody>
</table>

Table of races. (e.g. row 1: Horse number 9 won the pre-race and horse number 10 was second)

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Class</th>
<th>Type</th>
<th>Winner1</th>
<th>Winner2</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>Race</td>
<td>Pre-Race</td>
<td>Horse9</td>
<td>Horse10</td>
</tr>
<tr>
<td>65</td>
<td>Race</td>
<td>Race1</td>
<td>Horse2</td>
<td>Horse1</td>
</tr>
<tr>
<td>80</td>
<td>Race</td>
<td>Race2</td>
<td>Horse6</td>
<td>Horse8</td>
</tr>
<tr>
<td>99</td>
<td>Race</td>
<td>Race3</td>
<td>Horse1</td>
<td>Horse8</td>
</tr>
</tbody>
</table>
2.  
- Bets have to be placed before the beginning of the first race.
- Only accept bets that chose a horse of the given subset (e.g. the line-up of a race)
- See RETE graph

3.  
(P Evaluate_Bets
(Race
  ^Type <> Pre-Race )
(Race
  ^Type Race1
  ^Winner1 <W1> )
(Race
  ^Type Race2
  ^Winner1 <W2> )
(Race
  ^Type Race3
  ^Winner1 <W3> )
(Bet
  ^Tip1 <<< Horse1 Horse2 Horse3 Horse5 >>>
  ^Tip2 <<< Horse4 Horse6 Horse7 Horse8 >>>
  ^Tip3 <<< Horse1 Horse2 Horse4 Horse8 >>>
  ^Tip1 <W1>
  ^Tip2 <W2>
  ^Tip3 <W3>
  ^Timestamp < 65 )

→ ...

4.  
(P Evaluate_Tip1
(Race
  ^Type Race1
  ^Winner1 <w> )
(Bet
  ^Tip1 <w> )

→ ...

(P Evaluate_Tip2
(Race
  ^Type Race2
  ^Winner1 <w> )
(Bet
  ^Tip2 <w> )

→ ...

(P Evaluate_Tip3
(Race
  ^Type Race3
  ^Winner1 <w> )
(Bet
  ^Tip3 <w> )

→ ...
and so on for race2 and race3...

merged
Exercise 1: [Deduction - Wang] (3 Points)

a) Prove the following implication with Wang’s algorithm:

\[((\neg A \lor (B \land \neg C)) \land \neg D) \lor \neg (A \land C) \Rightarrow \neg A \lor \neg C\]

Exercise 2: [Deduction] (6 Points)

Prove each of the following theorems with both resolution and connection method:

a) Premises: (B ∨ C) (A ∨ ¬C) (¬A ∨ C) (¬A ∨ ¬C)
   Theorem: B

b) Premises: (C ⇒ (A ∨ B)) (A ∨ B ∨ C) (B ⇒ A)
   Theorem: A

Exercise 3: [Resolution, Green's Answer Predicate] (5 Points)

Formalize the following statements in predicate logic and answer the question: **Whom Adam loves?**

- Eva loves food.
- Eva or Anna love wine.
- Adam loves everybody, who loves wine.

Submission deadline is this Thursday (26th of May) before 16:00 at the chair.
Exercise 1: [Deduction - Wang] (3 Points)

a) Prove the following implication with Wang’s algorithm:

\[(\neg A \lor (B \land \neg C)) \land \neg D) \lor \neg (A \land C) \Rightarrow \neg A \lor \neg C\]

Solution:

\[(\neg A \lor (B \land \neg C)) \land \neg D) \lor \neg (A \land C) \Rightarrow \neg A \lor \neg C\]

Exercise 2: [Deduction] (6 Points)

Prove each of the following theorems with both resolution and connection method:

a) Premises: (B \lor C)  (A \lor \neg C)  (\neg A \lor C)  (\neg A \lor \neg C)  
Theorem: B

b)
Premises: \((C \Rightarrow (A \lor B))\) \((A \lor B \lor C)\) \((B \Rightarrow A)\)

Theorem: \(A\)

Solution:

a)
Resolution:
To be contradicted: \((B \lor C) \land (A \lor \neg C) \land (\neg A \lor \neg C) \land (\neg A \lor \neg C) \land (\neg B)\)

Connection method:
Disjunctive normal form: \((\neg B \land \neg C) \lor (\neg A \land C) \lor (A \land \neg C) \lor (\neg A \land C) \lor B\)

Set notation: \{{\{\neg B, \neg C\}\{\neg A, C\}\{A, \neg C\}\{A, C\}\{B\}}\}

Matrix Notation:

\[
\begin{array}{cccc}
\neg B & \neg A & A & A \\
& & & B \\
& & \neg C & C & \neg C & C \\
\end{array}
\]

Paths:
- \{\neg B, \neg A, A, A, B\}
- \{\neg B, \neg A, A, C, B\}
- \{\neg B, \neg A, \neg C, A, B\}
- \{\neg B, C, A, A, B\}
- \{\neg B, C, A, C, B\}
- \{\neg B, C, \neg C, A, B\}
- \{\neg B, C, \neg C, C, B\}
- \{\neg C, \neg A, A, A, B\}
- \{\neg C, \neg A, A, C, B\}
- \{\neg C, \neg A, \neg C, A, B\}

{¬C ¬A ¬C C B}
{¬C C A A B}
{¬C C A C B}

{¬C C ¬C A B}
{¬C C ¬C C B}

All paths include connections -> theorem is proven

b)

Resolution:
To be contradicted: (C ⇒ (A ∨ B)) ∧ (A ∨ B ∨ C) ∧ (B ⇒ A) ∧ (¬A)

\[ \neg C \lor A \lor B \]
\[ A \lor B \lor C \]
\[ \neg B \lor A \]
\[ \neg A \]

Connection method:

\((C \Rightarrow (A \lor B)) \land (A \lor B \lor C) \land (B \Rightarrow A) \land (\neg A)\)

\(A\)

Disjunctive normal form:

\((C \land \neg A \land \neg B) \lor (\neg A \land \neg B \land \neg C) \lor (B \land \neg A) \lor A\)

Set notation:

\{\{C, \neg A, \neg B\}, \{\neg A, \neg B, \neg C\}, \{B, \neg A\}, \{A\}\}

Matrix notation:

\[
\begin{array}{ccc}
C & \neg A & B \\
\neg A & \neg B & A \\
\neg B & \neg C & \neg A
\end{array}
\]

Paths:

C ¬A B A
C ¬A ¬A A
C ¬B B A
C ¬B ¬A A
C ¬C B A
C ¬C ¬A A
¬A ¬A B A
¬A ¬A ¬A A
Exercise 3: [Resolution, Green'sches Antwortprädikat] (5 Points)

Formalise the following statements in predicate logic and answer the question: Whom Adam loves?

- Eva loves Food
- Eva or Anna (or both) love wine
- Adam loves everybody, who loves wine

Solution:

\[\text{Eva loves food} \rightarrow \{\text{loves(Eva,Food)}\}\]
\[\text{Eva or Anna (or both) love wine} \rightarrow \{\text{loves(Eva,Wine) , loves(Anna,Wine)}\}\]
\[\text{Adam loves everybody, who loves wine} \rightarrow \text{loves(x,Wine) }\Rightarrow \text{loves(Adam,x) }\]
\[= \{\text{loves(Adam,x) , not loves(x,Wine)}\}\]

Whom Adam loves? \[\rightarrow \{\text{not loves(Adam,y) , Ans(y)}\}\]

Resolution of \{loves(Adam,x) , not loves(x,Wine)\} and \{not loves(Adam,y) , Ans(y)\} with \[y/x\] results in:
\[\{\text{not loves(y,Wine) , Ans(y) }\}\]

Resolution of \{loves(Eva,Wine) , loves(Anna,Wine)\} and \{not loves(y,Wine) , Ans(y) \} with \[Eva/y\] results in:
\[\{\text{loves (Anna,Wine), Ans(Eva) }\}\]

Resolution of \{loves(Anna,Wine), Ans(Eva) \} und \{not loves(y,Wein) , Ans(y) \} with \[Anna/y\] results in:
\[\{ \text{Ans(Anna), Ans(Eva) }\}\]

\[\Rightarrow \text{Adam loves Anna or Eva}\]
Exercise 1: [Default Theory - Basics] (5 Points)

1. Formulate the following statements in default logic:
   1) Normally the president of the USA is a man.
   2) Typically the German economy is growing.
   3) Typically Germans eat sauerkraut.

2. Express in your own words what the following default formulas express:
   a) \( \frac{\text{Student}(x) \land \text{AlwaysAttendsLecture}(x,y) \land \text{Comes}(x)}{\text{GetsManyPointsInTheExam}(x)} \)
   b) \( \frac{\text{Politician}(x) \land \text{Comes}(x) \land \neg \text{Comes}(y) \land \neg \text{Comes}(z)}{\text{EarnsMuchMoney}(x)} \)

Exercise 2: [Default Theory - Reasoning] (5 Points)

Write down all extensions of the following default theories \((D_i, W_i)\)

a) \( D_1 = \{ \text{Party} : \text{Comes}(Peter), \text{Party} : \text{Comes}(Mary) \} \)
   \( W_1 = \{ \text{Party}, \{ \text{Party} \land \text{Comes}(Peter) \} \rightarrow \neg \text{Comes}(Mary) \} \)

b) \( D_2 = \{ \text{Party} : \text{Comes}(Peter), \text{Party} : \text{Comes}(Mary), \text{Party} \land \text{Comes}(Peter) : \neg \text{Comes}(Mary) \} \)
   \( W_2 = \{ \text{Party} \} \)

b) \( D_3 = \{ \neg A(x) \} \)
   \( W_3 = \{ \} \)

d) \( D_4 = \{ \text{Party} : \text{Comes(Peter)} \land \text{Comes(Mary)} \} \)
   \( W_4 = \{ \text{Party} \} \)

e) \( D_5 = \{ \neg B \land \neg C \land \neg A \} \)
   \( W_5 = \{ \} \)

Submission deadline is this Wednesday (1st of June) before 16:00 at the chair.
Methods of Artificial Intelligence, SS 2011

Assignment 6 – Solution

Exercise 1: [Default Theory - Basics] (5 Points)

1. a) $\text{PresidentOfTheUSA}(x) \land \text{Man}(x) \Rightarrow \text{Man}(x)$

1. b) $\text{GermanEconomy}(x) \land \text{Growing}(x) \Rightarrow \text{Growing}(x)$

1. c) $\text{German}(x) \land \text{EatsSauerkraut}(x) \Rightarrow \text{EatsSauerkraut}(x)$

2. a) Typically a student who always attends a certain lecture y, gets many points in the exam.

   b) Normally a politician earns much money.

Exercise 2: [Default Theory - Reasoning] (5 Points)

a) $E_1 = \{ \text{Party}, (\text{Party} \land \text{Comes(Peter)}) \rightarrow \neg \text{Comes(Mary)}, \text{Comes(Mary)} \}$

b) $E_2 = \{ \text{Party}, (\text{Party} \land \text{Comes(Peter)}) \rightarrow \neg \text{Comes(Mary)}, \text{Comes(Peter)} \}$

b) $E_1 = \{ \text{Party}, \text{Comes(Peter)}, \text{Comes(Mary)} \}$

b) $E_2 = \{ \text{Party}, \text{Comes(Peter)}, \neg \text{Comes(Mary)} \}$

2. c) There is no extension for this default theory. One can argue that there is an extension and that this extension is empty. When looking at the algorithm however, it becomes clear that it will not terminate properly, so the correct solution would be that there is no extension.

d) $E_1 = \{ \text{Party}, \text{Comes(Peter)}, \neg \text{Comes(Mary)} \}$

e) There is also no extension for this default theory. Because of the circular dependencies no candidate is an extension.
Assignment 7

Exercise 1: [Certainty Factors] (10 Points)

The values displayed on top of the edges constitute the Certainty Factors of the according rule.

Which Certainty Factors would MYCIN apply to the conclusions E and F?

Submission deadline is this Thursday (9\textsuperscript{th} of June) before 16:00 at the chair.
Exercise 1: [Certainty Factors] (10 Points)

The values displayed on top of the edges constitute the Certainty Factors of the according rule.

Which Certainty Factors would MYCIN apply to the conclusions E and F?

Certainty Factors of the rules:
- \(\text{CF} [A \rightarrow C] = 0.5\)
- \(\text{CF} [B \rightarrow D] = 0.5\)
- \(\text{CF} [C \rightarrow D] = 1.0\)
- \(\text{CF} [C \rightarrow E] = -1.0\)
- \(\text{CF} [D \rightarrow E] = 0.8\)
- \(\text{CF} [E \rightarrow F] = 0.5\)

Node A:
- \(\text{CF} [A] = 1.0\)

Node B:
- \(\text{CF} [B] = 1.0\)

Node C:
- \(\text{CF} [C] = \text{CF} [C, A] = \text{CF} [A] \times \text{CF} [A \rightarrow C] = 1.0 \times 0.5 = 0.5\)

Node D:
- \(\text{CF} [D, B] = \text{CF} [B] \times \text{CF} [B \rightarrow D] = 0.5 \times 0.5 = 0.5\)
- \(\text{CF} [D, C] = \text{CF} [C] \times \text{CF} [C \rightarrow D] = 0.5 \times 1.0 = 0.5\)
- \(\text{CF} [D] = \text{CF} [D, B \land C] = \text{MB} [D, B \land C], \text{because}\)
  - \(\text{CF} [D, B] > 0\)
  - \(\text{CF} [D, C] > 0\) (positive evidences)
- \(\text{MB} [D, B \land C] = \text{MB} [D, B] + \text{MB} [D, C] - \text{MB} [D, B] \times \text{MB} [D, C], \text{where}\)
  - \(\text{MB} [D, B] = \max [0, \text{CF} [D, B]] = 0.5\)
  - \(\text{MB} [D, C] = \max [0, \text{CF} [D, C]] = 0.5\)
  - \(\Rightarrow \text{MB} [D, B \land C] = 0.5 + 0.5 - 0.5 \times 0.5 = 1.0 - 0.25 = 0.75 = \text{CF} [D]\)

Node E:
- \(\text{CF} [E, C] = \text{CF} [C] \times \text{CF} [C \rightarrow E] = 0.5 \times (-1.0) = -0.5\)
- \(\text{CF} [E, D] = \text{CF} [D] \times \text{CF} [D \rightarrow E] = 0.75 \times 0.8 = 0.6\)
- \(\text{CF} [E] = \frac{\text{CF} [E, C] + \text{CF} [E, D]}{1 - \min [\text{CF} [E, C], \text{CF} [E, D]]} = \frac{-0.5 + 0.6}{1 - \min [-0.5, 0.6]} = \frac{0.1}{1 - 0.5} = 0.1 = 0.2\)

Node F:
- \(\text{CF} [F, E] = \text{CF} [E] \times \text{CF} [E \rightarrow F] = 0.2 \times 0.5 = 0.1 = \text{CF} [F]\)
Exercise 1: [Searching - Graph Search] (5 Points)
Consider a search space where the start node is number 1 and the successors of a node \( n \) are the nodes \( 2n \) and \( 2n + 1 \).

1. Draw the search space for the nodes 1 to 15.
2. Suppose the goal node is 11. List the order in which nodes will be visited for breadth-first search and depth-first search (with limit 3). Don’t use the variation from Michie/Ross.

Exercise 2: [Searching - Graph Search] (10 Points)
You want to drive from Strasbourg to Munich in the shortest possible amount of time. From earlier rides you know how much time is needed for the routes between the different cities and you have noted them down in the map below. The numbers in brackets \((x, y)\) denote coordinates of the different cities. Use the A* search algorithm to find the fastest route between Strasbourg and Munich. Use the following heuristic:

\[
h(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2},
\]

where \( p_1 \) and \( p_2 \) are cities with coordinates \((x_1, y_1)\) and \((x_2, y_2)\) respectively.

Write down the order in which the nodes are expanded by drawing the search tree. Annotate each node with the values for \( f \), \( g \) and \( h \).
Exercise 3: [Searching - Graph Search] (10 Points)
Suppose that an agent is in a $3 \times 3$ maze environment like the one shown below. The agent knows that its initial location is (1, 1), that the goal is at (3, 3), and that the four actions *Up, Down, Left, Right* have their usual effects unless blocked by a wall (see below figure left). The agent does not know where the internal walls are (see below figure right). In any given state, the agent perceives the set of legal actions (it perceives the set of unblocked directions); it can also tell whether the state is one it has visited before or a new state.

1. How many possible environment configurations (combinations of walls) exist?
2. How many distinct percepts are possible in the initial state (see below figure right)?
3. Describe the first few branches of a graph for this problem (depth = 2). How large (roughly) is the complete graph?

![A simple maze problem. The agent starts at S and must reach G (left), but knows nothing of the environment (right).](image)

Submission deadline is this Thursday (16th of June) before 16:00 at the chair.

Please notice:
The tutorials for Assignment 7 (in the week 13th June to 19th June) are cancelled and will be held together with the tutorial for Assignment 8 (in the week 20th June to 26th June).

On 4th July 2011 the Monday tutorial group, 10:00-12:00, will be held in Building C2 2, Room 24.
Exercise 1: [Searching - Graph Search] (5 Points)

Consider a search space where the start node is number 1 and the successors of a node $n$ are the nodes $2n$ and $2n + 1$.

1. Draw the search space for the nodes 1 to 15.
   Solution: See the figure below.

2. Suppose the goal node is 11. List the order in which nodes will be visited for breadth-first search and depth-first search (with limit 3). Don’t use the variation from Michie/Ross.
   Solution:
   
   Breadth-first search: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
   Depth-first search: 1, 2, 4, 8, 9, 5, 10, 11
Exercise 2: [Searching - Graph Search] (10 Points)
You want to drive from Strasbourg to Munich in the shortest possible amount of time. From earlier rides you know how much time is needed for the routes between the different cities and you have noted them down in the map below. The numbers in brackets (x, y) denote coordinates of the different cities. Use the A* search algorithm to find the fastest route between Strasbourg and Munich. Use the following heuristic:
\[ h(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \]
where \( p_1 \) and \( p_2 \) are cities with coordinates (\( x_1, y_1 \)) and (\( x_2, y_2 \)) respectively.
Write down the order in which the nodes are expanded by drawing the search tree. Annotate each node with the values for \( f \), \( g \) and \( h \).
Exercise 3: [Searching - Graph Search] (10 Points)
Suppose that an agent is in a 3 × 3 maze environment like the one shown below. The agent knows that its initial location is (1, 1), that the goal is at (3, 3), and that the four actions Up, Down, Left, Right have their usual effects unless blocked by a wall (see below figure left). The agent does not know where the internal walls are (see below figure right). In any given state, the agent perceives the set of legal actions (it perceives the set of unblocked directions); it can also tell whether the state is one it has visited before or a new state.

1. How many possible environment configurations (combinations of walls) exist?
   Solution: There are 12 possible locations for internal walls, so there are $2^{12} = 4096$ possible environment configurations. Before seeing any percepts all 4096 configurations are possible.

2. How many distinct percepts are possible in the initial state (see below figure right)?
   Solution: Assuming the external walls are known, there are two internal walls and hence $2^2 = 4$ possible percepts.

   ![Initial State and Percept Diagram]
   A simple maze problem. The agent starts at S and must reach G (left), but knows nothing of the environment (right).

3. Describe the first few branches of a graph for this problem (depth = 2).
   Solution: See the figure below.
   How large (roughly) is the complete graph?
   Solution: The first null action leads to four possible belief states, as shown in the figure below. From each belief state, the agent chooses a single action which can lead to up to 8 belief states (on entering the middle square). Given the possibility of having to retrace its steps at a dead end, the agent can explore the entire maze in no more than 18 steps, so the complete plan (expressed as a tree) has no more than $8^{18}$ nodes. On the other hand, there are just $3^{12}$ possible nodes (there are 12 possible locations for internal walls, wall ∈ {unknown, no, yes}) and not all of them are reachable.

   ![Graph Diagram]
Exercise 1: [Minimax, Alpha-Beta] (6 Points)

Given are the following tree of a 2-person game and the ratings of game situations after 3 moves at the leaves of the tree.

1) Determine the moves that player A should chose in order to get the best possible rating according to the Minimax method.

2) Which steps could be cut down while rating the inner nodes by applying Alpha-Beta pruning instead of a complete Minimax rating?
Exercise 2: [RMS] (9 Points)
Have a look at the following dependency network.

1) Using the STATUS function, mark the network in such a manner that it is complete and consistent.
2) Label the nodes of the network according to the function rank.
3) Draw the support tree for believe node U9.
4) Determine the following sets:
   a) All assumption justifications
   b) All assumptions
   c) All premise justifications
   d) All premises
   e) Current support set CS (U9)
   f) Current support set CS (U7)
Exercise 3: [Model Based Diagnosis] (9 Points)
Have a look at the following electric circuit.

1) Give a system description (SD) for the component type $S$ in the form of predicates.
2) Write down the observations („val“) of the measurements.
3) Fill in the missing values in the diagnosis table below:

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td></td>
</tr>
<tr>
<td>$c = 12$</td>
<td></td>
</tr>
<tr>
<td>$d = -2$</td>
<td></td>
</tr>
<tr>
<td>$e = -14$</td>
<td></td>
</tr>
<tr>
<td>$f = -19$</td>
<td></td>
</tr>
<tr>
<td>$g = 7$</td>
<td></td>
</tr>
</tbody>
</table>

Submission deadline is this Wednesday (22nd of June) before 16:00 at the chair.

Please notice:
On 4th July 2011 the Monday tutorial group, 10:00-12:00, will be held in Building C2 2, Room 24.
Exercise 1: [Minimax, Alpha-Beta] (6 Points)

Given are the following tree of a 2-person game and the ratings of game situations after 3 moves at the leaves of the tree.

1) Determine the moves that player A should chose in order to get the best possible rating according to the Minimax method.

2) Which steps could be cut down while rating the inner nodes by applying Alpha-Beta pruning instead of a complete Minimax rating?

Solution:

1) Player A should choose the move in the middle (marked by red arrow).

2) The underlined nodes are pruned away.
Exercise 2: [RMS] (9 Points)

Have a look at the following dependency network.

1) Using the STATUS function, mark the network in such a manner that it is complete and consistent.
   Solution: See the figure below.

2) Label the nodes of the network according to the function rank.
   Solution: See the green numbers in the figure below.

3) Draw the support tree for believe node U9.
   Solution: There are two possibilities: See the orange or the red edges in the figure below.

4) Determine the following sets:
   a) All assumption justifications
      Solution: \{R7\}
   b) All assumptions
      Solution: \emptyset
   c) All premise justifications
      Solution: \{R1, R3, R5\}
   d) All premises
      Solution: \{U1, U4, U5, U9\}
   e) Current support set CS(U9)
      Solution: CS(U9) = \{U4, U5, U7, U8\}
   f) Current support set CS(U7)
      Solution: CS(U7) = \{U1, U6, U3\}
Exercise 3: [Model Based Diagnosis] (9 Points)
Have a look at the following electric circuit.

1) Give a system description (SD) for the component type $S$ in the form of predicates.

   **Sample Solution:**
   
   \[
   \text{type}(S, \text{Subtractor}) \land \text{ok}(S) \land \text{val}(\text{in1}(S), V1) \land \text{val}(\text{in2}(S), V2) \land V3 = V2 - V1 \rightarrow \text{val}(\text{out}(S), V3) \\
   \text{type}(S, \text{Subtractor}) \land \text{ok}(S) \land \text{val}(\text{out}(S), V3) \land \text{val}(\text{in2}(S), V2) \land V1 = V2 - V3 \rightarrow \text{val}(\text{in1}(S), V1) \\
   \text{type}(S, \text{Subtractor}) \land \text{ok}(S) \land \text{val}(\text{out}(S), V3) \land \text{val}(\text{in1}(S), V1) \land V2 = V1 + V3 \rightarrow \text{val}(\text{in2}(S), V2)
   \]

2) Write down the observations („val“) of the measurements.

   **Sample Solution:**
   
   \[
   \text{val}(a, 5), \text{val}(b, 2), \text{val}(c, 12), \text{val}(d, -2), \text{val}(e, -14), \text{val}(f, -19), \text{val}(g, 7)
   \]

3) Fill in the missing values in the diagnosis table below:

   **Sample Solution:**
   
<table>
<thead>
<tr>
<th>Measurement</th>
<th>Diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>$c = 12$</td>
<td>$[[S1],[S2],[A1],[A2],[A3]]$</td>
</tr>
<tr>
<td>$d = -2$</td>
<td>$[[S1],[A1], [A2,S2],[A3]]$</td>
</tr>
<tr>
<td>$e = -14$</td>
<td>$[[A1], [A2,S2],[A3]]$</td>
</tr>
<tr>
<td>$f = -19$</td>
<td>$[[A2,S2],[A3]]$</td>
</tr>
<tr>
<td>$g = 7$</td>
<td>$[[A2,S2]]$</td>
</tr>
</tbody>
</table>

   **Legende:**
   
   - $\text{in1}$
   - $\text{in2}$
   - $\text{in1} - \text{in2}$
   - $\text{in2} + \text{in1}$
Exercise 1: [STRIPS] (4+5+3 Points)
STRIPS has originally been developed for a robot called “Shakey”. In the figure below, you can see a version of the Shakey world. It consists of four rooms along a corridor. Each room has a door and a light switch. The extended actions in the Shakey world are: going from position to position, push movable object (e.g. crates), climb on and off solid objects (e.g. crates) and switch the light on or off.
(a) Describe the following actions by specifying the preconditions and the resulting effects (ADD-formulas):

1. Go(x, y)
   It is necessary that Shakey is at Position x and that x and y are positions in the same room. (Note: Doors are positions that belong to the adjoining room as well as to the corridor)

2. Push(b, x, y)
   Moving of the Crate b from Position x to Position y in the same room.

3. ClimbUp(b) and ClimbDown(b)
   Climp on Crate b or climb down from Crate b respectively.

4. TurnOn(s) and TurnOff(s)
   Turn a Switch s on or off.
   In order to reach the switch, Shakey has to stand on a crate that is at the position of the switch.

(b) Describe the initial state as it is depicted in the figure above.
   (Note: Switch 1 and Switch 4 are switched on)

(c) Construct a plan by using the actions from part (a), in which Shakey pushes Crate 2 in Room 2.

Exercise 2: [STRIPS] (10 Points)
The monkey-and-bananas problem is faced by a monkey in a laboratory with some bananas hanging out of reach from the ceiling. A box is available that will enable the monkey to reach the bananas if he climbs on it. Initially, the monkey is at A, the bananas at B, and the box at C. The monkey and box have height Low, but if the monkey climbs onto the box he will have height High, the same as the bananas. The actions available to the monkey include Go from one place to another, Push an object from one place to another, ClimbUp onto or ClimbDown from an object, and Grasp or Ungrasp an object. Grasping results in holding the object if the monkey and object are in the same place at the same height.

(a) Write down the initial state description.

(b) Write down STRIPS-style definitions of the six actions.

Submission deadline is this Thursday (30th of June) before 16:00 at the chair.

Due to some questions we want to point out again our admission requirements for the exam:

- Registration (Course Website + HISPOS)
- Attendance of all tutorials
- Demonstration of one exercise solution in a tutorial session
- At least 50% of available points of the exercises

Please notice:

On 4th July 2011 the Monday tutorial group, 10:00-12:00, will be held in Building C2 2, Room 24.
Exercise 1: [STRIPS] (4+5+3 Points)
STRIPS has originally been developed for a robot called “Shakey”. In the figure below, you can see a version of the Shakey world. It consists of four rooms along a corridor. Each room has a door and a light switch. The extended actions in the Shakey world are: going from position to position, push movable object (e.g. crates), climb on and off solid objects (e.g. crates) and switch the light on or off.

(a) Describe the following actions by specifying the preconditions and the resulting effects (ADD-formulas):

1. **Go(x, y)**
   - It is necessary that Shakey is at Position x and that x and y are positions in the same room. (Note: Doors are positions that belong to the adjoining room as well as to the corridor)

2. **Push(b, x, y)**
   - Moving of the Crate b from Position x to Position y in the same room.

3. **ClimbUp(b) and ClimbDown(b)**
   - Climp on Crate b or climb down from Crate b respectively.

4. **TurnOn(s) and TurnOff(s)**
   - Turn a Switch s on or off.
   - In order to reach the switch, Shakey has to stand on a crate that is at the position of the switch.
Sample Solution:

(1) \textit{Go}(x, y)
- \textbf{PRE:} \text{At}(\text{Shakey}, x), \text{In}(x, r), \text{In}(y, r), \text{On}(\text{Shakey}, \text{Floor})
- \textbf{DEL:} \text{At}(\text{Shakey}, x)
- \textbf{ADD:} \text{At}(\text{Shakey}, y)

(2) \textit{Push}(b, x, y)
- \textbf{PRE:} \text{At}(\text{Shakey}, x), \text{Pushable}(b), \text{At}(b, x), \text{On}(\text{Shakey}, \text{Floor})
- \textbf{DEL:} \text{At}(b, x), \text{At}(\text{Shakey}, x)
- \textbf{ADD:} \text{At}(b, y), \text{At}(\text{Shakey}, y)

(3) \textit{ClimbUp}(b)
- \textbf{PRE:} \text{At}(\text{Shakey}, x), \text{At}(b, x), \text{Climbable}(b), \text{On}(\text{Shakey}, \text{Floor})
- \textbf{DEL:} \text{On}(\text{Shakey}, b)

\textit{ClimbDown}(b)
- \textbf{PRE:} \text{On}(\text{Shakey}, b)
- \textbf{DEL:} \text{On}(\text{Shakey}, b)
- \textbf{ADD:} \text{On}(\text{Shakey}, \text{Floor})

(4) \textit{TurnOn}(s)
- \textbf{PRE:} \text{On}(\text{Shakey}, b), \text{At}(\text{Shakey}, x), \text{At}(s, x), \text{TurnedOff}(s)
- \textbf{DEL:} \text{TurnedOff}(s)
- \textbf{ADD:} \text{TurnedOn}(s)

\textit{TurnOff}(s)
- \textbf{PRE:} \text{On}(\text{Shakey}, b), \text{At}(\text{Shakey}, x), \text{At}(s, x), \text{TurnedOn}(s)
- \textbf{DEL:} \text{TurnedOn}(s)
- \textbf{ADD:} \text{TurnedOff}(s)

(b) Describe the initial state as it is depicted in the figure above.
(Note: Switch 1 and Switch 4 are switched on)

Sample Solution:
\text{In}(\text{Switch1}, \text{Room1}) \land \text{In}(\text{Door1}, \text{Room1}) \land \text{In}(\text{Door1}, \text{Corridor})
\text{In}(\text{Switch2}, \text{Room2}) \land \text{In}(\text{Door2}, \text{Room2}) \land \text{In}(\text{Door2}, \text{Corridor})
\text{In}(\text{Switch3}, \text{Room3}) \land \text{In}(\text{Door3}, \text{Room3}) \land \text{In}(\text{Door3}, \text{Corridor})
\text{In}(\text{Switch4}, \text{Room4}) \land \text{In}(\text{Door4}, \text{Room4}) \land \text{In}(\text{Door4}, \text{Corridor})
\text{In}(\text{Shakey}, \text{Room3}) \land \text{At}(\text{Shakey}, \text{XS}) \land \text{On}(\text{Shakey}, \text{Floor})
\text{In}(\text{Crate1}, \text{Room1}) \land \text{In}(\text{Crate2}, \text{Room1}) \land \text{In}(\text{Crate3}, \text{Room1}) \land \text{In}(\text{Crate4}, \text{Room1})
\text{Climbable(Crate1)} \land \text{Climbable(Crate2)} \land \text{Climbable(Crate3)} \land \text{Climbable(Crate4)}
\text{Pushable(Crate1)} \land \text{Pushable(Crate2)} \land \text{Pushable(Crate3)} \land \text{Pushable(Crate4)}
\text{At(Crate1, X1)} \land \text{At(Crate2, X2)} \land \text{At(Crate3, X3)} \land \text{At(Crate4, X4)}
\text{At(Schalter1, X1)} \land \text{At(Schalter2, X2)} \land \text{At(Schalter3, X3)} \land \text{At(Schalter4, X4)}
\text{TurnedOn(Switch1)} \land \text{TurnedOn(Switch4)} \land \text{TurnedOff(Switch2)} \land \text{TurnedOff(Switch3)}

(c) Construct a plan by using the actions from part (a), in which Shakey pushes Crate 2 in Room 2.

Sample Solution:
\text{Go}(\text{XS}, \text{Door3})
\text{Go}(\text{Door3}, \text{Door1})
\text{Go}(\text{Door1}, \text{X2})
\text{Push(Crate2, X2, Door1)}
\text{Push(Crate2, Door1, Door2)}
\text{Push(Crate2, Door2, Switch2)}
Exercise 2: [STRIPS] (10 Points)
The monkey-and-bananas problem is faced by a monkey in a laboratory with some bananas hanging out of reach from the ceiling. A box is available that will enable the monkey to reach the bananas if he climbs on it. Initially, the monkey is at A, the bananas at B, and the box at C. The monkey and box have height Low, but if the monkey climbs onto the box he will have height High, the same as the bananas. The actions available to the monkey include Go from one place to another, Push an object from one place to another, ClimbUp onto or ClimbDown from an object, and Grasp or Ungrasp an object. Grasping results in holding the object if the monkey and object are in the same place at the same height.

(a) Write down the initial state description.

Sample Solution:
\[ At(Monkey, A) \land At(Bananas, B) \land At(Box, C) \land Height(Monkey, Low) \land Height(Box, Low) \land Height(Bananas High) \land Pushable(Box) \land Climbable(Box) \]

(b) Write down STRIPS-style definitions of the six actions.

Sample Solution:
(5) Go(x, y)
  PRE: At(Monkey, x)
  DEL: At(Monkey, x)
  ADD: At(Monkey, y)

(6) Push(b, x, y)
  PRE: At(Monkey, x), Pushable(b), At(b, x)
  DEL: At(b, x), At(Monkey, x)
  ADD: At(b, y), At(Monkey, y)

(7) ClimbUp(b)
  PRE: At(Monkey, x), At(b, x), Climbable(b), Height(Monkey, Low)
  DEL: Height(Monkey, Low)
  ADD: Height(Monkey, High)

ClimbDown(b)
  PRE: Height(Monkey, High)
  DEL: Height(Monkey, High)
  ADD: Height(Monkey, Low)

(8) Grasp(b)
  PRE: Height(Monkey, h), Height(b, h), At(Monkey, x), At(b, x)
  DEL: At(b, x), Height(b, h)
  ADD: Holding(b)

Ungrasp(b)
  PRE: Holding(b), At(Monkey, x), Height(Monkey, h)
  DEL: Holding(b)
  ADD: At(b, x), Height(b, h)