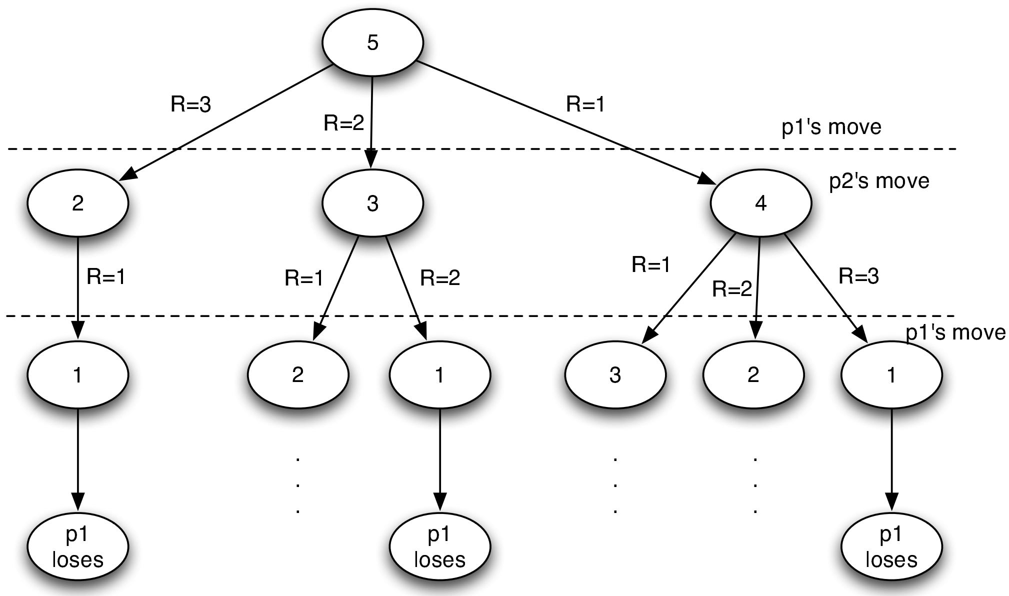
2nd Midterm Exam Key

Wednesday April 15
75 minutes == 75 points
Open book and notes - no computer

1. 15 points - *The game of nim is played as follows. There is a stack of 5 pennies. Two players alternate removing 1, 2, or 3 pennies from the stack. The player who picks up the last penny loses. Show by drawing the game graph that the player who plays the second move can always win.*

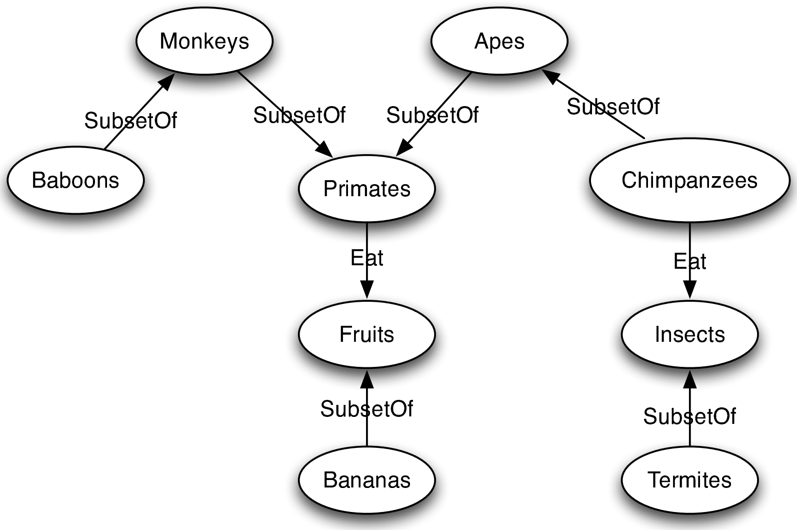


In the above graph, the states are represented by the number of pennies left in the stack. With his first move, the player with the second turn (p2) can choose to force the player going first (p1) into taking the last penny (i.e., always choose to remove a number R of pennies to leave only one for the first player's second turn.) p2 can do this regardless of the number of pennies p1 removes in his first move.

2. 15 points

   (a) *Represent the following set of facts in a semantic network:*
   Apes and monkeys are primates.
   All primates eat fruits.
   Chimpanzees are apes.
   Baboons are monkeys.

Chimpanzees eat insects.
Bananas are fruits.
Termites are insects.



(b) *What procedure could you use to answer the following kind of question: "Does A eat B?". Describe it informally in terms of how the arcs in the networks will be traversed.*

Here, we want to employ a sort of bidirectional graph search. Start from the most specialized object "A" (e.g., Chimpanzees or Baboons). Check if it has a property "eats". If it has this property check if "A" eats "B". If not, move up to the next set in the inheritance tree, and check for the "eats" property. For example, if we started from "Baboon", we'd next check "Monkeys". Continue this until either a superset containing "A" has the property "eats" with relation to "B" or all of the sets "A" inherits from are traversed. Note that doing this will only take us as far as "eats fruit". When we start with "A", we also need to start with "B" and similar to a bidirectional search, we need to move up the hierarchy to see if the path from "A" and from "B" intersect. Alternatively, once we find the property "eats", we have to go down the tree to see if "B" is in that tree. For instance "Bananas" is a subclass of fruit, so when we find "eats fruit", we have to see if "B" is a subclass of "Fruit"

(c) *Answer the following questions and explain briefly your reasoning:*

Does a baboon eat bananas?

Yes. "Baboon" inherits from "Monkeys" which inherits from "Primates" which eat "Fruits". "Bananas" are a subset of "Fruits"

Does a baboon eat termites?

No. "Baboons" inherit nothing from "Chimpanzees", which are the only subset of "Primates" on the network that eat termites.

3. 15 points - *All people who are not poor and are smart are happy. Those people who read are smart. John can read and is not poor. Happy people have exciting lives. Can anyone be found with an exciting life?*

   (a) *Write the sentences in predicate calculus, using appropriate predicates.*

      i. $\forall x \neg Poor(x) \wedge Smart(x) \Rightarrow Happy(x)$
      ii. $\forall x Reads(x) \Rightarrow Smart(x)$
      iii. $Reads(John) \wedge \neg Poor(John)$
      iv. $\forall x Happy(x) \Rightarrow HasExcitingLife(x)$
      v. $\exists x HasExcitingLife(x)$

   (b) *Convert them to Conjunctive Normal Form.*

      i. $Poor(x) \vee \neg Smart(x) \vee Happy(x)$
      ii. $\neg Reads(y) \vee Smart(y)$
      iii. $Reads(John)$
      iv. $\neg Poor(John)$
      v. $\neg Happy(z) \vee HasExcitingLife(z)$
      vi. $HasExcitingLife(S)$

   (c) *Who has an exciting life? Answer by using resolution with refutation.*

   First, we're trying to prove that someone has an exciting life, $\exists x HasExcitingLife(x)$. We negate, convert to CNF, and add it to the KB. This becomes $\neg HasExcitingLife(w)$. Since we are trying to answer a question, we can use an answer literal. We take a literal, like $Answer(personWithExcitingLife)$, and add it to the negated goal via disjunction. So our negated goal becomes: $\neg HasExcitingLife(w) \vee Answer(w)$. Now, this resolution process generates an answer whenever a clause is produced containing only a single answer literal. Thus, our KB is:

      i. $Poor(x) \vee \neg Smart(x) \vee Happy(x)$
      ii. $\neg Reads(y) \vee Smart(y)$
      iii. $Reads(John)$
      iv. $\neg Poor(John)$
      v. $\neg Happy(z) \vee HasExcitingLife(z)$
      vi. $\neg HasExcitingLife(w) \vee Answer(w)$
      We then apply resolution to arrive at a single answer literal ($Answer$(???)).
      vii. $\neg Smart(John) \vee Happy(John)$ $\boxed{\text{(i,iv) } \{x/John\}}$
      viii. $Smart(John)$ $\boxed{\text{(ii,iii) } \{y/John\}}$
      ix. $Happy(John)$ $\boxed{\text{(vii,viii)}}$
      x. $HasExcitingLife(John)$ $\boxed{\text{(v,ix) } \{z/John\}}$
      xi. $Answer(John)$ $\boxed{\text{(vi,x) } \{w/John\}}$

4. 15 points - *Write the following sentences in predicate calculus, using appropriate predicates:*

   (a) All big houses are expensive.
   $\forall x[House(x) \wedge Big(x)] \Rightarrow Expensive(x)$

   (b) A house is prestigious only if it is big.
   $\forall x[House(x) \wedge Prestigious(x)] \Rightarrow Big(x)$
   Note that "p only if q" gets translated as $p \Rightarrow q$, and that "only if" is **not** the same as "if and only if".

   (c) Any small apartment costs less than any big house.
   $\forall x, y[Apartment(x) \wedge House(y) \wedge Small(x) \wedge Big(y)] \Rightarrow Cost(x) < Cost(y)$

   (d) There is a house which is bigger than any apartment.
   $\exists h \forall a House(h) \wedge Apartment(a) \Rightarrow Size(h) > Size(a)$

   (e) All apartments have at least one bathroom.
   $\forall a Apartment(a) \Rightarrow \exists b Bathroom(b) \wedge In(b, a)$

   (f) There is only one red house.
   $\exists h House(h) \wedge Red(h) \wedge \forall x House(x) \wedge Red(x) \Rightarrow x = y$

5. 15 points - *Answer the following questions briefly but precisely. Justify your answers.*

   (a) *Suppose you use resolution to prove that $KB \models \alpha$. Does is mean that α is valid?*
   A valid sentence is one that is true in all models. If $KB \models \alpha$, then we know that α is true in all models for which the KB is true. However, this does not include all models, so no, this does not mean that α is valid.

   (b) *Is it true that in first-order logic, if a sentence is entailed, it can always be proven using resolution with refutation?*

   Yes. This is because resolution is refutation-complete, meaning that if a set of sentences is unsatisfiable, then resoluton will always be able to derive a contradiction. That is, if $KB \models \alpha$, then $KB \wedge \neg\alpha$ can always be resolved to NIL.

   (c) *Is it true that it is always possible to prove that a sentence in propositional logic is entailed or not entailed by a knowledge base? Is this also true if the sentence is in predicate calculus?*

   Because resolution is refutation-complete (as noted in the last question), we can always prove entailment of a sentence. The question is if we can disprove entailment.

   In propositional logic, we can do this. This is because propositonal logic is decidable. This means, with propositional logic, we can always decide if a given sentence has membership (or not) with a valid set of sentences in the KB.

   However, in first-order logic, we can't do this. This is because first-order logic is semidecidable. We can only decide if a sentence is entailed, but not if it isn't entailed.