ZOOKEEPER -- A Deduction System to Identify Animals in a Zoo

Z1 If ?x has hair
    Then ?x is a mammal

Z2 If ?x gives milk
    Then ?x is a mammal

Z3 If ?x has feathers
    Then ?x is a bird

Z4 If ?x flies
    ?x lays eggs
    Then ?x is a bird

Z5 If ?x is a mammal
    ?x eats meat
    Then ?x is a carnivore

Z6 If ?x is a mammal
    ?x has pointed teeth
    ?x has claws
    ?x has forward-pointing eyes
    Then ?x is a carnivore

Z7 If ?x is a mammal
    ?x has hoofs
    Then ?x is an ungulate

Z8 If ?x is a mammal
    ?x chews cud
    Then ?x is an ungulate

Z9 If ?x is a carnivore
    ?x has tawny color
    ?x has dark spots
    Then ?x is a cheetah

Z10 If ?x is a carnivore
    ?x has tawny color
    ?x has black strips
    Then ?x is a tiger

Z11 If ?x is an ungulate
    ?x has long legs
    ?x has long neck
    ?x is tawny color
    ?x has dark spots
    Then ?x is a giraffe

Z12 If ?x is a mammal
    ?x has white color
    ?x has black stripes
    Then ?x is a zebra

Z13 If ?x is a bird
    ?x does not fly
    ?x has long legs
    ?x has long neck
    ?x is black and white
    Then ?x is an ostrich

Z14 If ?x is a bird
    ?x does not fly
    ?x swims
    ?x is black and white
    Then ?x is a penguin

Z15 If ?x is a bird
    ?x is a good flyer
    Then ?x is an albatross

Stretch has hair.
Stretch chews cud.
Stretch has long legs.
Stretch has a long neck.
Stretch has tawny color.
Stretch has dark spots.
To forward chain (detailed version),

> Until no rule produces a new assertion,

> For each rule,

> Try to match the first antecedent with an existing assertion. Create a new binding set with variable bindings established by the match.

> Using the existing variable bindings, try to match the next antecedent with an existing assertion. If any new variables appear in this antecedent, augment the existing variable bindings.

> Repeat the previous step for each antecedent, accumulating variable bindings as you go, until,

> There is no match with any existing assertion using the binding set established so far. In this case, back up to a previous match of an antecedent to an assertion, looking for an alternative match that produces an alternative, workable binding set.

> There are no more antecedents to be matched. In this case,

> Use the binding set in hand to instantiate the consequent.

> Determine if the instantiated consequent is already asserted. If not, assert it.

> Back up to the most recent match with unexplored bindings, looking for an alternative match that produces a workable binding set.

> There are no more alternatives matches to be explored at any level.

To backward chain,

> Find a rule whose consequent matches the hypothesis (or antecedent) and create a binding set (or augment the existing binding set).

> Using the existing binding set, look for a way to deal with the first antecedent,

> Try to match the antecedent with an existing assertion.

> Treat the antecedent as an hypotheses and try to support it by backward chaining through other rules using the existing binding set.

> Repeat the previous step for each antecedent, accumulating variable bindings, until,

> There is no match with any existing assertion or rule consequent using the binding set established so far. In this case, back up to the most recent match with unexplored bindings, looking for an alternative match that produces a workable binding set.

> There are no more antecedents to be matched. In this case, the binding set in hand supports the original hypothesis.

> If all possible binding sets are desired, report the current binding set, and back up, as if there were no match.

> If only one possible binding set is desired, report the current binding set and quit.

> There are no more alternative matches to be explored at any level.
BAGGER – A Rule-based Reaction System

1. Check-order step
   (suggest additions)
2. Bag-large-items step
   (bag big bottles first)
3. Bag-medium-items step
   (put frozen items in freezer bags)
4. Bag-small-items step

Working Memory:

Current Step is check-order.
Bag1 is bag.
Bread is to be bagged.
Glop is to be bagged.
Granola is to be bagged.
Ice cream is to be bagged.
Potato chips are to be bagged.

B1 If Current step is check-order
    Potato chips are to be bagged
    There is no Pepsi to be bagged
    Then Ask the customer whether he would like a bottle of Pepsi

B2 If Current step is check-order
    Then Current step is no longer check-order
    Step is bag-large-items

Or

B2 If Current step is check-order
    Delete Current step is check-order
    Add Current step is bag-large items

B3 If Current step is bag-large-items
    A large item is to be bagged
    The large item is a bottle
    The current bag contains < 6 large items
    Delete The large item is to be bagged
    Add The large item is in the current bag
B4  If  Current step is bag-large-items
    A large item is to be bagged
    The current bag contains < 6 large items
Delete  The large item is to be bagged
Add    The large item is in the current bag

B5  If  Current step is bag-large-items
    A large item is to be bagged
    An empty bag is available
Delete  The current bag is the current bag
Add    The empty bag is the current bag

B6  If  Current step is bag-large-items
Delete  Current step is bag-large-items
Add    Current step is bag-medium-items

B7  If  Current step is bag-medium-items
    A medium item is frozen, but not in a freezer bag
Delete  The medium item is not in a freezer bag
Add    The medium item is in a freezer bag

B8  If  Current step is bag-medium-items
    A medium item is to be bagged
    The current bag is empty or contains only medium items
    The current bag contains no large items
    The current bag contains < 12 medium items
Delete  The medium item is to be bagged
Add    The medium item is in the current bag

B9  If  Current step is bag-medium-items
    A medium item is to be bagged
    An empty bag is available
Delete  The current bag is the current bag
Add    The empty bag is the current bag

B10 If  Current step is bag-medium-items
Delete  Current step is bag-medium-items
Add    Current step is bag-small-items

B11 If  Current step is bag-small-items
    A small item is to be bagged
    The current bag contains no large items
    The current bag contains no medium item
    The current bag contains < 18 small item
Delete  The small item is to be bagged
Add    The small item is in the current bag

B12 If  Current step is bag-small-items
    A small item is to be bagged
    An empty bag is available
Delete  The current bag is the current bag
Add    The empty bag is the current bag

B13 If  Current step is bag-small-items
Delete  Current step is bag-small-items
Add    Current step is “done”