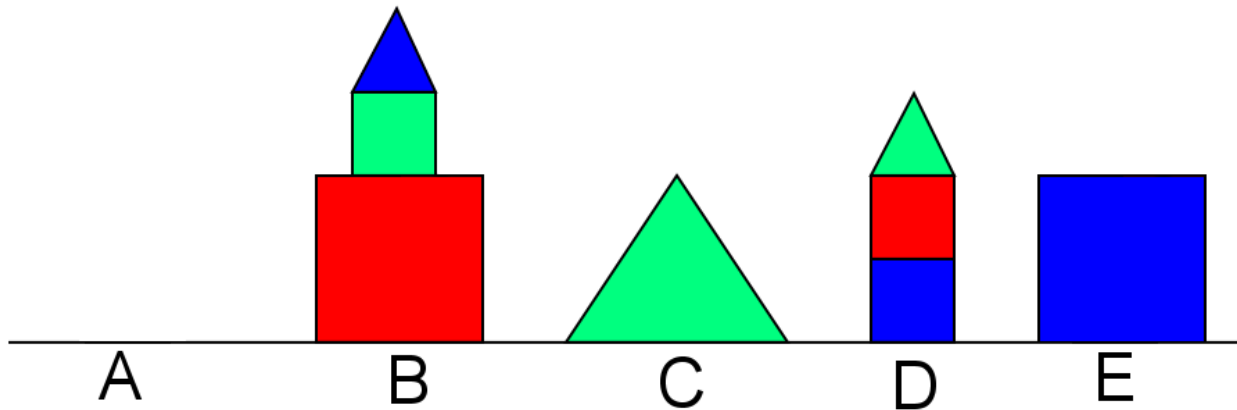


Blocks World: Work with your partner(s) to understand and write Prolog dynamic predicates to perform actions in the blocks world.

Return to the following world containing blocks.



- The blocks can be cubes or pyramids.
- The blocks can be located either on the table or on another block.
- Each block is on at most one other block (which must be a cube) and has at most one other block directly on it.
- The table is divided into five contiguous areas, A,B,C,D, and E from left to right.
- Each block on the table is located in exactly one of these areas, and each table area has at most one block directly on it.

Follow the instructions:

1. Copy the Prolog parser which is in file `ch8_4.pl` as well as some utilities contained in `wordUtils.pl`. Load these files as well as the blocks world and lexicon files that you wrote previously. The parser contains two predicates “ask” and “act”. The ask predicate takes a sentence as uses the np parser from before. You can test it with any noun phrases such as

`ask('the block on the big red cube').`

2. In your world model, write an additional relation `putOn(X,Y)` which will change the world by moving block X to be stacked on top of block Y, if possible. To model moving a block within the blocks world you will need to use the dynamic predicates *assert* and *retract*.

For example, if we want to change a block X that was previously on block Y and now have it on block Z would first have to indicate that the *on* relation which takes two arguments can be changed and then retract the fact that X is on Y and assert the fact that X is now on Z with:

```
:-dynamic on/2
myNewPred(X,Z) :- on(X,Y), retract(on(X,Y)), assert(on(X,Z)), fail.
```

The fail at the end forces all the clauses to be evaluated.

Complete the putOn relation keeping in mind that a block has to be at the top of a stack to be moved onto another block that also has to be on top of a stack. You probably need another relation in your world model to keep track of whether a block is on top of a stack. Also, remember that you can't stack a block on top of a pyramid.

You may test your work by using the "act" predicate in the parser and then check which blocks on are other blocks with an on(X,Y) query.

3. In your world model write an additional relation moveTo(X,Y) which will move block X to an empty location Y. For example, in the world above assuming that the block is named smallBluePyramid and the location is named locA we could then use moveTo(smallBluePyramid,locA) to take the blue pyramid off the first stack and move it to the empty location A.

You may want an additional relation in your world to indicate whether a location is empty.

Write an additional "act" rule for dealing with sentences like "move the small blue pyramid to locA".