//Simulate learning - suppose action gamble\_bindle\_0\_0 was tried.

ActionInstance currAct = util.getActionWithName(problem, "gamble\_bindle\_0\_0");

util.printIncompleteActionInstance(currAct);

//State of problem was (at\_0\_0). The action's pre is met.

//The action is executed. Now, state == (at\_0\_0 ^ have\_bindle).

//We have learned that "have\_bindle" IS NOT a possible add effect (UNLISTEDEFFECT).

//We have learned that "have\_bindle" IS an add effect.

//We must encode this new knowledge in the BDD.

//The risk associated with this knowledge piece is:

Risk risk = Risk.getRiskFromIndex(Risk.UNLISTEDEFFECT, currAct.getName(), "have\_bindle");

//Prove we got the risk:

System.out.println(risk.toString());

//We insert this knowledge into the BDD

util.learnedIntoBDD(risk, true);

//Let's query the BDD to see what it says about this risk.

System.out.println("\* " + util.bdd.and(util.riskToBDD.get(risk), util.bddRef));//130

System.out.println("\* " + util.bdd.and(util.bdd.not(util.riskToBDD.get(risk)), util.bddRef));//0

System.out.println();

//Simulate learning - Action move\_0\_0\_1\_0 was tried.

currAct = util.getActionWithName(problem, "move\_0\_0\_1\_0");

util.printIncompleteActionInstance(currAct);

//State of problem was (at\_0\_0 ^ have\_bindle). The action's pre is met.

//The action is executed. Now, state == (at\_1\_0 ^ have\_bindle).

//We have learned that "have\_bindle" IS NOT a possible delete effect (PossClob).

//We have learned that "have\_bindle" IS NOT a delete effect.

//We must encode this new knowledge in the BDD.

//The risk associated with this knowledge piece is:

Risk risk2 = Risk.getRiskFromIndex(Risk.POSSCLOB, currAct.getName(), "have\_bindle");

//Prove we got the risk:

System.out.println(risk2.toString());

//We insert this knowledge into the BDD

util.learnedIntoBDD(risk2, false);

//Let's query the BDD to see what it says about this risk.

System.out.println("\* " + util.bdd.and(util.riskToBDD.get(risk2), util.bddRef));//0

System.out.println("\* " + util.bdd.and(util.bdd.not(util.riskToBDD.get(risk2)), util.bddRef));//162

System.out.println();

//Let's get an action to set up up a query about a risk that will not be loaded into the BDD

currAct = util.getActionWithName(problem, "move\_0\_0\_0\_1");

util.printIncompleteActionInstance(currAct);

//Let's setup up a query about a risk that is not in the BDD

Risk risk3 = Risk.getRiskFromIndex(Risk.POSSCLOB, currAct.getName(), "have\_bindle");

//Let's query the BDD to see what it says about this risk.

System.out.println("\* " + util.bdd.and(util.riskToBDD.get(risk3), util.bddRef));//164

System.out.println("\* " + util.bdd.and(util.bdd.not(util.riskToBDD.get(risk3)), util.bddRef));//166

System.out.println();

//Seems like 0 is the only value worth seeking.

//Now, how to encode possPre combinations?

//Then, how to update the action model:

// Should remove the learned props from the BDD - though we could just leave them there

//For single props learned only

void learnedIntoBDD(Risk risk, boolean isTrue){

Integer reference = riskToBDD.get(risk);

if(isTrue){

Integer temp = bdd.ref(bdd.and(bddRef, reference));

bdd.deref(bddRef);

bddRef = temp;

}else{

Integer temp = bdd.ref(bdd.and(bddRef, bdd.not(reference)));

bdd.deref(bddRef);

bddRef = temp;

}

}