

# Ejercicio: Sistemas Basados en Lógica

Carlos Sanabria Miranda

```
public static FOLDomain simpsonsExerciseDomain() {
    FOLDomain domain = new FOLDomain();
    // Constants
    domain.addConstant("Juan");
    domain.addConstant("Simpsons");
    // Predicates
    domain.addPredicate("DibujoAnimado");
    domain.addPredicate("Infantil");
    domain.addPredicate("ParaAdultos");
    domain.addPredicate("Ninio");
    domain.addPredicate("Entender");
    domain.addPredicate("Ver");
    domain.addPredicate("Aburrirse");
    return domain;
}

public static FOLKnowledgeBase createSimpsonsExerciseKnowledgeBase(
    InferenceProcedure infp) {
    FOLKnowledgeBase kb = new FOLKnowledgeBase(DomainFactory.simpsonsExerciseDomain(),
    infp);

    // Algunos dibujos animados no son infantiles
    kb.tell("EXISTS x (DibujoAnimado(x) AND NOT Infantil(x))");
    // Los dibujos animados no infantiles son para adultos
    kb.tell("FORALL x ( (DibujoAnimado(x) AND NOT Infantil(x) ) => ParaAdultos(x) )");
    // Los niños que ven dibujos animados para adultos no los entienden y se aburren
    kb.tell("FORALL x FORALL y ( (Ninio(x) AND DibujoAnimado(y) AND ParaAdultos(y) AND Ver(x,y)
    )"
        + " => (NOT Entender(x,y) AND Aburrirse(x)) )");
    // Los Simpsons son dibujos animados para adultos
    kb.tell("(DibujoAnimado(Simpsons) AND ParaAdultos(Simpsons))");
    // Juan es un niño y ve los Simpsons
    kb.tell("(Ninio(Juan) AND Ver(Juan,Simpsons))");

    return kb;
}

private static void simpsonsExerciseDemo(InferenceProcedure ip) {
    StandardizeApartIndexicalFactory.flush();

    FOLKnowledgeBase kb = FOLKnowledgeBaseFactory
        .createSimpsonsExerciseKnowledgeBase(ip);
```

```

String kbStr = kb.toString();

// ¿Existe algún niño que se aburra?
String query = "EXISTS x (Ninio(x) AND Aburrirse(x))";

InferenceResult answer = kb.ask(query);

System.out.println("\nSimpsons Exercise Knowledge Base:");
System.out.println(kbStr);
System.out.println("Query: " + query);
for (Proof p : answer.getProofs()) {
    System.out.print(ProofPrinter.printProof(p));
    System.out.println("");
}
}

Simpsons Exercise Knowledge Base:
EXISTS x (DibujoAnimado(x) AND NOT(Infantil(x)))
FORALL x ((DibujoAnimado(x) AND NOT(Infantil(x))) => ParaAdultos(x))
FORALL x FORALL y (((Ninio(x) AND DibujoAnimado(y)) AND ParaAdultos(y)) AND Ver(x,y)) => (NOT(Entender(x,y)) AND Aburrirse(x)))
(DibujoAnimado(Simpsons) AND ParaAdultos(Simpsons))
(Ninio(Juan) AND Ver(Juan,Simpsons))

Query: EXISTS x (Ninio(x) AND Aburrirse(x))
Proof, Answer Bindings: {x=Juan}
-----
|Step | Proof
-----
|1 | (DibujoAnimado(Simpsons) AND ParaAdultos(Simpsons))
|2 | (DibujoAnimado(Simpsons) AND ParaAdultos(Simpsons))
|3 | (Ninio(Juan) AND Ver(Juan,Simpsons))
|4 | (Ninio(Juan) AND Ver(Juan,Simpsons))
|5 | FORALL x FORALL y (((Ninio(x) AND DibujoAnimado(y)) AND ParaAdultos(y)) AND Ver(x,y)) => (NOT(Entender(x,y)) AND Aburrirse(x)))
|6 | [-Aburrirse(v10), ~Ninio(v10), Answer0(v10)]
|7 | [DibujoAnimado(Simpsons)]
|8 | [ParaAdultos(Simpsons)]
|9 | [Ninio(Juan)]
|10 | [Ver(Juan,Simpsons)]
|11 | [-DibujoAnimado(y), ~Ninio(x), ~ParaAdultos(y), ~Ver(x,y), Aburrirse(x)]
|12 | [-DibujoAnimado(v4), ~Ninio(v3), ~ParaAdultos(v4), ~Ver(v3,v4), Aburrirse(v3)]
|13 | [-DibujoAnimado(v9), ~Ninio(v8), ~ParaAdultos(v9), ~Ver(v8,v9), Aburrirse(v8)]
|14 | [-DibujoAnimado(c17), ~Ninio(c18), ~ParaAdultos(c17), ~Ver(c18,c17), Answer0(c18)]
|15 | [-DibujoAnimado(Simpsons), ~Ninio(Juan), ~ParaAdultos(Simpsons), Answer0(Juan)]
|16 | [-DibujoAnimado(Simpsons), ~ParaAdultos(Simpsons), Answer0(Juan)]
|17 | [-DibujoAnimado(Simpsons), Answer0(Juan)]
|18 | [Answer0(Juan)]
-----

```