Chapter 2

A2: Logics

2.0.1 Taller

```
% problem:
% James is taller than Kate and Carly.
%Sammy is shorter than Kate.
%Natalie is shorter than Kate and Sammy,
%however Sammy is shorter than Carly.
%Who is the shortest?

assign(max_models, -1). %all possible models % assign(max_models,1). -> one m
assign(domain_size,5).
%taller(x, y) -> x is taller than y
%shorter(x, y) -> x is shorter than y

list(distinct).
 [James, Kate, Sammy, Natalie, Carly].
end_of_list.

formulas(taller).
        shorter(x, y) <-> taller(y, x).
        -shorter(x,x).
        -taller(x,x).
        taller(x, y) -> -shorter(x, y).
        shorter(x, y)-> -taller(x, y).

        taller(x ,y) & shorter(z, y) -> taller(x, z).

        taller(James, Kate).
        taller(James, Carly).
        shorter(Sammy, Kate).
        shorter(Natalie, Kate).
        shorter(Natalie, Sammy).
        shorter(Sammy, Carly).

        % exists x taller(Natalie, x).
```

RESULTS:

```
bia@DESKTOP-FPO7FLO:~/Prover9/bin$ ./mace4 -c -f A2/tallerFolder.in
============================== Mace4 =================================
Mace4 (64) version 2017-11A (CIIRC), November 2017.
Process 460 was started by bia on DESKTOP-FPO7FLO,
Sun Dec 24 16:25:03 2023
The command was "./mace4 -c -f A2/tallerFolder.in".
============================== end of head ===========================
============================== INPUT =================================
% Reading from file A2/tallerFolder.in
assign(max_models,-1).
assign(domain_size,5).
% assign(domain_size, 5) -> assign(start_size, 5).
% assign(domain_size, 5) -> assign(end_size, 5).
list(distinct).
[James,Kate,Sammy,Natalie,Carly].
end_of_list.
formulas(taller).
shorter(x,y) <-> taller(y,x).
-shorter(x,x).
-taller(x,x).
taller(x,y) -> -shorter(x,y).
shorter(x,y) -> -taller(x,y).
taller(x,y) & shorter(z,y) -> taller(x,z).
taller(James,Kate).
taller(James,Carly).
shorter(Sammy,Kate).
shorter(Natalie,Kate).
shorter(Natalie,Sammy).
shorter(Sammy,Carly).
end_of_list.
============================== end of input ==========================
============================== PROCESS NON-CLAUSAL FORMULAS ==========
% Formulas that are not ordinary clauses:
1 shorter(x,y) <-> taller(y,x) # label(non_clause).  [assumption].
2 taller(x,y) -> -shorter(x,y) # label(non_clause).  [assumption].
3 shorter(x,y) -> -taller(x,y) # label(non_clause).  [assumption].
4 taller(x,y) & shorter(z,y) -> taller(x,z) # label(non_clause).  [assumption
============================== end of process non-clausal formulas ===
============================== CLAUSES FOR SEARCH ====================
formulas(mace4_clauses).
-shorter(x,y) | taller(y,x).
shorter(x,y) | -taller(y,x).
-shorter(x,x).
-taller(x,x).
-taller(x,y) | -shorter(x,y).
-shorter(x,y) | -taller(x,y).
-taller(x,y) | -shorter(z,y) | taller(x,z).
taller(James,Kate).
taller(James,Carly).
shorter(Sammy,Kate).
```

```
shorter(Natalie,Kate).
shorter(Natalie,Sammy).
shorter(Sammy,Carly).
James != Kate.
James != Sammy.
James != Natalie.
James != Carly.
Kate != Sammy.
Kate != Natalie.
Kate != Carly.
Sammy != Natalie.
Sammy != Carly.
Natalie != Carly.
end_of_list.
========== end of clauses for search ==========
% There are no natural numbers in the input.
========== DOMAIN SIZE 5 ==========
==== Mace4 starting on domain size 5. ====
========== MODEL ==========
interpretation( 5, [number=1, seconds=0], [
function(Carly, [ 0 ]),
function(James, [ 1 ]),
function(Kate, [ 2 ]),
function(Natalie, [ 3 ]),
function(Sammy, [ 4 ]),
relation(shorter(_,_), [
0, 1, 0, 0, 0,
0, 0, 0, 0, 0,
0, 1, 0, 0, 0,
1, 1, 1, 0, 1,
1, 1, 1, 0, 0 ]),
relation(taller(_,_), [
0, 0, 0, 1, 1,
1, 0, 1, 1, 1,
0, 0, 0, 1, 1,
0, 0, 0, 0, 0,
0, 0, 0, 1, 0 ])
]).
========== end of model ==========
========== MODEL ==========
interpretation( 5, [number=2, seconds=0], [
function(Carly, [ 0 ]),
function(James, [ 1 ]),
function(Kate, [ 2 ]),
function(Natalie, [ 3 ]),
function(Sammy, [ 4 ]),
relation(shorter(_,_), [
0, 1, 0, 0, 0,
0, 0, 0, 0, 0,
1, 1, 0, 0, 0,
```

```
1, 1, 1, 0, 1,
1, 1, 1, 0, 0 ]),
relation(taller(_,_), [
0, 0, 1, 1, 1,
1, 0, 1, 1, 1,
0, 0, 0, 1, 1,
0, 0, 0, 0, 0,
0, 0, 0, 1, 0 ])
]).
========== end of model ==========
========== MODEL ==========
interpretation( 5, [number=3, seconds=0], [
function(Carly, [ 0 ]),
function(James, [ 1 ]),
function(Kate, [ 2 ]),
function(Natalie, [ 3 ]),
function(Sammy, [ 4 ]),
relation(shorter(_,_), [
0, 1, 1, 0, 0,
0, 0, 0, 0, 0,
0, 1, 0, 0, 0,
1, 1, 1, 0, 1,
1, 1, 1, 0, 0 ]),
relation(taller(_,_), [
0, 0, 0, 1, 1,
1, 0, 1, 1, 1,
1, 0, 0, 1, 1,
0, 0, 0, 0, 0,
0, 0, 0, 1, 0 ])
]).
========== end of model ==========
========== STATISTICS ==========
For domain size 5.
Current CPU time: 0.00 seconds (total CPU time: 0.01 seconds).
Ground clauses: seen=251, kept=191.
Selections=6, assignments=14, propagations=53, current_models=3.
Rewrite_terms=46, rewrite_bools=325, indexes=6.
Rules_from_neg_clauses=1, cross_offs=10.
========== end of statistics ==========
User_CPU=0.01, System_CPU=0.00, Wall_clock=0.
Exiting with 3 models.
------ process 460 exit (all_models) ------
Process 460 exit (all_models) Sun Dec 24 16:25:03 2023
The process finished Sun Dec 24 16:25:03 2023
end_of_list.
```

2.0.2 threeYoungWomen

```
%There are three young newly married women,
%and each has a daughter.
```

```
%Determine whose daughter is whose, and how old each child is.
%Women: Sara, Katherine, and Mallory
%Daughters: Abby, Jessica, and Eliana
%Ages: 4, 5, and 6

%Clues:
%1. Sara and her daughter have lunch once a week with Katherine and her daugl
%2. Jessica's mother doesn't like talking to Sara and her husband.
%3. Sara's daughter is 6.

assign(domain_size,3). % 0 is age4, 1 is age 5, 2 is age 6
assign(max_models, -1).
set(arithmetic).


list(distinct).
        [Abby, Jessica, Eliana].
        [Sara, Katherine, Mallory].
end_of_list.

formulas(three).
        daughter(x) = y & daughter(z) = y -> x = z.

        daughter(0) < daughter(1).
        daughter(1) < daughter(2).

        daughter(Katherine) = Eliana.
        Eliana = 1.

        daughter(Sara) != Jessica.

        daughter(Sara) = 2.

end_of_list.
```

RESULT:

```
========== MODEL ==========
interpretation( 3, [number=1, seconds=0], [
function(Eliana, [ 1 ]),
function(Jessica, [ 0 ]),
function(Katherine, [ 1 ]),
function(Sara, [ 2 ]),
function(daughter(_), [ 0, 1, 2 ]),
function(Abby, [ 2 ]),
function(Mallory, [ 0 ])
]).
========== end of model ==========
========== STATISTICS ==========
For domain size 3.
Current CPU time: 0.00 seconds (total CPU time: 0.00 seconds).
```

```
Ground clauses: seen=39, kept=30.
Selections=9, assignments=27, propagations=37, current_models=1.
Rewrite_terms=270, rewrite_bools=170, indexes=18.
Rules_from_neg_clauses=18, cross_offs=44.
============================== end of statistics =============================
User_CPU=0.00, System_CPU=0.00, Wall_clock=0.
Exiting with 1 model.
------- process 468 exit (all_models) -------
Process 468 exit (all_models) Sun Dec 24 16:27:09 2023
The process finished Sun Dec 24 16:27:09 2023
```

2.0.3 twoCubeCalendar

Problem from the lab. 2 dice, write on them number so the date can be read.

```
    assign(max_models,-1).
assign(domain_size,9). %6 can be used for 9
%all possible models
set(arithmetic).

list(distinct).
        [f0 , f1 , f2, f3 ,f4 ,f5].
        [g0, g1, g2, g3, g4, g5].
end_of_list.

formulas(assumptions) .

        f0 = 0.
        f1 = 1.
        f2 = 2.
        f3 = 3.
        f0 < f1 & f1< f2 &  f2<f3  & f3< f4 &  f4< f5.

        g0 = 0.
        g1 = 1.
        g2 = 2.

        g0 < g1 & g1 < g2 & g2 < g3 & g3 < g4 & g4 <g5.

        f3 < g3.
        g3 < f4.
        f4 < g4.
        g4 < f5.
        f5 < g5.
        f3 != g3.
        f4 != g4.
        f5 != g5.

        %(f0 = 0) & (f1 = 1) & (f2 = 2) & (f3 =  3) & (f4 = 4) & (f5 = 5).
        %(g0 = 0) & (g1 = 1) & (g2 = 2) & (g3 = 6) & (g4 = 7) & (g5 = 8).
```

```
end_of_list .

RESULTS:
========================================== MODEL ==========================================
interpretation( 9, [number=1, seconds=0], [
function(f0, [ 0 ]),
function(f1, [ 1 ]),
function(f2, [ 2 ]),
function(f3, [ 3 ]),
function(f4, [ 5 ]),
function(f5, [ 7 ]),
function(g0, [ 0 ]),
function(g1, [ 1 ]),
function(g2, [ 2 ]),
function(g3, [ 4 ]),
function(g4, [ 6 ]),
function(g5, [ 8 ])
]).
========================================== end of model ==========================================
========================================== STATISTICS ==========================================
For domain size 9.
Current CPU time: 0.00 seconds (total CPU time: 0.00 seconds).
Ground clauses: seen=55, kept=41.
Selections=31, assignments=279, propagations=7, current_models=1.
Rewrite_terms=1612, rewrite_bools=1238, indexes=0.
Rules_from_neg_clauses=0, cross_offs=188.
========================================== end of statistics ==========================================
User_CPU=0.00, System_CPU=0.01, Wall_clock=0.
Exiting with 1 model.
------ process 482 exit (all_models) ------
```

2.0.4 snailRace

```
% https://www.braingle.com/brainteasers/50710/snail-races.html
%People have gathered from around the world to witness the 2015 snail races.
%Each snail was handpicked by its country to race its way across
%the brutal 10cm track. Try to figure out the finishing time,
%country of origin, and shell color for each of the five snails.
%Don't be discouraged if it takes a while to solve,
%this is a snail race after all.

%1. Of Todd and the snail with the teal shell,
%one finished in 24 minutes and the other is from Mexico.

%2. The snail from Mexico finished 3 minutes after
%the snail with the lime shell, who finished 1 minute before
%the snail with the silver shell (who wasn't Hank).

%3. The snail from USA finished 2 minutes before Kipp,
%who finished some time before the snail from Spain.
```

```
%4. Of Ralph and the Spain contestant one had the fastest time and the other

%5. The snail from Canada was either the one with the teal shell or the one w

%answer:
%Hank, 26, Spain, violet
%Mike, 23, China, silver
%Kipp, 24, Canada, teal
%Ralph, 22, USA, lime
%Todd, 25, Mexico, grey

assign(domain_size,5).
assign(max_models, -1).
set(arithmetic).

list(distinct).
        [Hank, Mike, Kipp, Ralph, Todd].
        [Spain, China, Canada, USA, Mexico].
        [Violet, Silver, Teal, Lime, Grey].
        [T22, T23, T24, T25, T26].
end_of_list.

formulas(snailRace).
        %color(x) = y & color(z) = y -> x = z.
        T22 = 0.
        T23 = 1.
        T24 = 2.
        T25 = 3.
        T26 = 4.

% hint 1.
        (Todd = T24 & Teal = Mexico) | (Todd = Mexico & Teal = T24).

% hint 2.
        Hank != Silver.
        Lime + 3 = Mexico.
    Lime + 1 = Silver.

% hint 3.
        USA + 2 = Kipp.
        Kipp < Spain.

% hint 4.
        (Ralph = T22 & Spain = Violet) | (Spain = T22 & Ralph = Violet).

% hint 5.
        Canada = Teal | Canada = T25.
end_of_list.
```

RESULTS:

```
========== MODEL ==========
interpretation( 5, [number=1, seconds=0], [
function(Canada, [ 2 ]),
function(Hank, [ 4 ]),
function(Kipp, [ 2 ]),
function(Lime, [ 0 ]),
function(Mexico, [ 3 ]),
function(Ralph, [ 0 ]),
function(Silver, [ 1 ]),
function(Spain, [ 4 ]),
function(T22, [ 0 ]),
function(T23, [ 1 ]),
function(T24, [ 2 ]),
function(T25, [ 3 ]),
function(T26, [ 4 ]),
function(Teal, [ 2 ]),
function(Todd, [ 3 ]),
function(USA, [ 0 ]),
function(Violet, [ 4 ]),
function(China, [ 1 ]),
function(Grey, [ 3 ]),
function(Mike, [ 1 ])
]).
========== end of model ==========
========== STATISTICS ==========
For domain size 5.
Current CPU time: 0.00 seconds (total CPU time: 0.01 seconds).
Ground clauses: seen=59, kept=49.
Selections=293, assignments=1465, propagations=379, current_models=1.
Rewrite_terms=7993, rewrite_bools=4145, indexes=0.
Rules_from_neg_clauses=108, cross_offs=2554.
========== end of statistics ==========
User_CPU=0.01, System_CPU=0.00, Wall_clock=0.
Exiting with 1 model.
------ process 488 exit (all_models) ------
```

2.0.5 CopsAndThief

```
    %Three people met at a corner of a street.
%They all are dressed like cops, so they
%don't know who's the thief (bad guy).
%The cops will tell the truth (because they are good),
%and the thief will tell the truth too to
%make himself appear like a good cop.
%They are A, B and C. And they say this:

%A:"C's not the thief."
%B:"One of you both is the thief!"
```

```
%C:"I'm not the thief."

%Using this information, find out who the thief is.

%Answer
%A's the thief. He says C's not the thief,
%and B says one of the other two is,
%which means he must not be. So the only one left is A.

assign(domain_size,2).
assign(max_models, -1).

formulas(cops_thief).
        all x (human(x) -> cop(x) | thief(x)).
        all x ((cop(x) -> -thief(x)) & (thief(x) -> -cop(x))).
        cop(x) -> m(x).
        thief(x) -> m(x).
end_of_list.

formulas(puzzle).
        human(a) & human(b) & human(c).
        m(a) <-> -thief(c).
        m(b) <-> thief(a) | thief(c).
        m(c) <-> -thief(c).

end_of_list.
```

RESULTS:

```
========== MODEL ==========
interpretation( 2, [number=1, seconds=0], [
function(a, [ 0 ]),
function(b, [ 0 ]),
function(c, [ 1 ]),
relation(cop(_), [ 0, 1 ]),
relation(human(_), [ 1, 1 ]),
relation(m(_), [ 1, 1 ]),
relation(thief(_), [ 1, 0 ])
]).
========== end of model ==========
========== MODEL ==========
interpretation( 2, [number=2, seconds=0], [
function(a, [ 0 ]),
function(b, [ 1 ]),
function(c, [ 1 ]),
relation(cop(_), [ 0, 1 ]),
relation(human(_), [ 1, 1 ]),
relation(m(_), [ 1, 1 ]),
relation(thief(_), [ 1, 0 ])
]).
```

```
========== end of model ==========
========== STATISTICS ==========
For domain size 2.
Current CPU time: 0.00 seconds (total CPU time: 0.00 seconds).
Ground clauses: seen=18, kept=18.
Selections=8, assignments=15, propagations=32, current_models=2.
Rewrite_terms=49, rewrite_bools=126, indexes=45.
Rules_from_neg_clauses=0, cross_offs=0.
========== end of statistics ==========
User_CPU=0.00, System_CPU=0.00, Wall_clock=0.
Exiting with 2 models.
```

2.0.6 Secret Santa

```
% A group of about twenty friends decide to exchange gifts
% as secret Santas. Each person writes their name
% on a piece of paper and puts it in a hat and then
% each person randomly draws a name from the hat to
% determine who has them as their secret Santa.

% What is the probability that at least one person
% draws their own name?

%for domanin size = 5, nr generated models is 120

assign(domain_size,5).
assign(max_models,-1).
set(arithmetic).

list(distinct).
        [P0, P1, P2, P3, P4].%, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14,
end_of_list.

formulas(probability).
        %P0 = 0 | P0 = 1 | P0 = 2 | P0 = 3 | P0 = 4 | P0 = 5 | P0 = 6 | P0 =
        %P1 = 0 | P1 = 1 | P1 = 2 | P1 = 3 | P1 = 4 | P1 = 5 | P1 = 6 | P1 =

endend_of_list.

RESULT:
Exiting with 120 models.

Favorable Cases:
% A group of about twenty friends decide to exchange gifts
% as secret Santas. Each person writes their name
% on a piece of paper and puts it in a hat and then
% each person randomly draws a name from the hat to
% determine who has them as their secret Santa.

% What is the probability that at least one person
```

```
% draws their own name?

%results:
% for domain size  = 5, 76 generated models


assign ( domain_size ,5).
assign ( max_models , -1).
set(arithmetic).

list(distinct).
        [P0, P1, P2, P3, P4].%, P5, P6, P7, P8, P9, P10, P11, P12, P13, P14,
end_of_list.

formulas(probability).
        P0 = 0 | P1 = 1 | P2 = 2 | P3 = 3 | P4 = 4.
endend_of_list.

RESULTS:
Exiting with 76 models.

Probability: 76/120.

Same problem using functions:
Total Cases: 120 models.
assign ( domain_size ,5).
assign ( max_models , -1).
set(arithmetic).

list(distinct).
        [f(0), f(1), f(2), f(3), f(4)].
end_of_list.

formulas(probability).
        exists x (f(x) = x).
endend_of_list.

Favorable cases: (120 models generated, not good result)
assign ( domain_size ,5).
assign ( max_models , -1).
set(arithmetic).

list(distinct).
        [f(0), f(1), f(2), f(3), f(4)].
end_of_list.

formulas(probability).
        exists x (f(x) = x).
   f(0) = 0 | f(1) = 1 | f(2) = 2 | f(3) = 3 | f(4) = 4.
endend_of_list.
```