Homework Set 6

Michael Anderson

November 18, 2010

8.3 Each constant symbol in c may be mapped to any of the objects in the domain D, so there are D^c possible sets of mappings.

k-ary relations contain a set permutations of size k pulled from the set of constant objects, giving P_k^c such permutations, each of which either is or is not contained within a given relation. Thus, there are $2^{P_k^c}$ possible k-ary relations. Since there are p_k many orthogonal k-ary relations, there are $\prod_{k=1}^{A} 2^{(P_k^c)(p_k)}$ many possible sets of relations.

In a k-ary function each of the P_k^c tuples is mapped to one of the c constant symbols or the invisible object, giving $(P_k^c)(c+1)$ possible functions. There are then $\prod_{k=1}^{A}[(P_k^c)(c+1)]^{p_k}$ many possible sets of functions.

Since the mappings of constant objects, the functions, and the relations are all orthogonal to each other, the total number of possible models is:

$$(D^c)(\prod_{k=1}^{A} 2^{(P_k^c)(p_k)})(\prod_{k=1}^{A}[(P_k^c)(c+1)]^{p_k})$$

8.9 a)
  i. 2, because function arguments can only contain terms.
  ii. 1
  iii. 3, because this can be true if only one of Paris or Marseilles are in France.

b)
  i. 1
  ii. 3, because it is always true if c can take non-country values.
  iii. 3, because it is always true if c can take non-country values.
  iv. 2, because function arguments can only contain terms.

c)
  i. 3, because it is always false if c can take non-country values.
  ii. 1
  iii. 3, because it is false even if c is ever not a country, and is also not in South America.
  iv. 3, because it is always false if c can take non-country values.

d)
  i. 1
  ii. 1
  iii. 3, because it is true if some regions in Europe border a region in South America.
  iv. 1

e)
  i. 1
  ii. 1
  iii. 3, because it is false if x or y are not countries, or if they share a border, or they share a map color.

iv. 2, because the grammar does not contain $\neq$.

8.10 a) $Occupation(Emily, Surgeon) \lor Occupation(Emily, Lawyer)$

b) $Occupation(Joe, Actor) \land [\exists o\ \neg(o = Actor) \land Occupation(Joe, o)]$

c) $\forall p\ Occupation(p, Surgeon) \Rightarrow Occupation(p, Doctor)$

d) $\neg[\exists p\ Occupation(p, Lawyer) \land Customer(Joe, p)]$

e) $\exists p\ Boss(p, Emily) \land Occupation(p, Lawyer)$

f) $\exists p1\ [\forall p2\ Customer(p2, p1) \Rightarrow Occupation(p2, Doctor)]$

g) $\forall p1\ Occupation(p1, Surgeon) \Rightarrow [\exists p2\ Customer(p1, p2) \land Occupation(p2, Lawyer)]$

8.13 a)

$$\forall s\ Breezy(s) \Rightarrow \exists r\ Adjacent(r, s) \land Pit(r)$$

$$\forall s\ \neg[Breezy(s)] \Rightarrow \neg[\exists r\ Adjacent(r, s) \land Pit(r)]$$

Now reduce the conjunction to Equation (8.4):

$$(\forall s\ Breezy(s) \Rightarrow \exists r\ Adjacent(r, s) \land Pit(r)) \land (\forall s\ \neg[Breezy(s)] \Rightarrow \neg[\exists r\ Adjacent(r, s) \land Pit(r)])$$

$$(\forall s\ Breezy(s) \Rightarrow \exists r\ Adjacent(r, s) \land Pit(r)) \land (\forall s\ Breezy(s) \lor \neg[\exists r\ Adjacent(r, s) \land Pit(r)])$$

$$(\forall s\ Breezy(s) \Rightarrow \exists r\ Adjacent(r, s) \land Pit(r)) \land (\forall s\ \neg[\exists r\ Adjacent(r, s) \land Pit(r)] \lor Breezy(s))$$

$$(\forall s\ Breezy(s) \Rightarrow \exists r\ Adjacent(r, s) \land Pit(r)) \land (\forall s\ \exists r\ Adjacent(r, s) \land Pit(r) \Rightarrow Breezy(s))$$

$$\forall s\ Breezy(s) \Leftrightarrow \exists r\ Adjacent(r, s) \land Pit(r)$$

b)

$$\forall s\ Pit(s) \Rightarrow [\forall r\ Adjacent(r, s) \Rightarrow Breezy(r)]$$

This is insufficient, because we are only saying that a pit causes breezes in adjacent squares. Equation (8.4) also implies that a breeze causes a pit in one or more adjacent squares. Thus the following axiom is also necessary:

$$\forall s\ Breezy(s) \Rightarrow \exists r\ Adjacent(r, s) \land Pit(r)$$

8.20 a)

$$\exists y\ \neg(y \times (1 + 1) < x) \land \neg(x < y \times (1 + 1))$$

b)

$$\neg(\exists y, z\ (1 < y) \land (y < x) \land (\neg[(y \times z) < x] \land \neg[x < (y \times z)]))$$

c)

$$\forall x\ [\exists y\ \neg(y \times (1 + 1) < x) \land \neg(x < y \times (1 + 1))] \Rightarrow$$

$$[\exists p1, p2 (\neg(\exists y1, z1\ (1 < y1) \land (y1 < p1) \land (\neg[(y1 \times z1) < p1] \land \neg[p1 < (y1 \times z1)]))) \land$$

$$(\neg(\exists y2, z2\ (1 < y2) \land (y2 < p2) \land (\neg[(y2 \times z2) < p2] \land \neg[p2 < (y2 \times z2)]))) \land$$

$$\neg[(p1 + p2) < x] \land \neg[x < (p1 + p2)]]$$

8.24 $Student(p)$: Predicate. Person p is a student.
$Spring2001(c)$: Preciate. Class c was offered in Spring 2001.
$Took(p, c)$: Predicate. Person p took class c.
$Passed(p, c)$: Predicate. Person p passed class c.
$Score(c)$: Function. Returns the best score in class c.
$Higher(s1, s2)$: Predicate. Score s1 is higher than score s2.
$Buy(p, o)$: Predicate. Person p buys policy o.
$Smart(p)$: Predicate. Person p is smart.
$Expensive(o)$: Predicate. Policy o is expensive.
$Job(p, j)$: Predicate. Person p has job j.
$Sell(p1, p2)$ Predicate. Person p1 sells policies to person p2.
$Insured(p)$: Predicate. Person p is insured.
$ManInTown(p)$: Predicate. Person p is a man in town.
$Shaves(p1, p2)$: Predicate. Person p1 shaves person p2.
$UKBorn(p)$: Predicate. Person p was born in the UK.
$UKbyBirth(p)$: Predicate. Person p is a UK citizen by birth.
$UKbyDescent(p)$: Predicate. Person p is a UK citizen by descent.
$UKResident(p)$: Predicate. Person p is a UK Resident.
$Father(p)$: Function. Returns the father of p. $Mother(p)$: Function. Returns the mother of p. $Fool(p1, p2, t)$ Predicate. Person p1 can fool person p2 at time t.
$Speaks(x, l)$ Predicate. Person x speaks language l.
$French, Greek$: Constants denoting classes.
$Agent, Barber, Politician$: Constants denoting jobs. $Greek_person$: Constant denoting an ethnicity.

a) $\exists p\ Student(p) \land Took(p, French) \land Spring2001(French)$

b) $\forall p\ [Student(p) \land Took(p, French)] \Rightarrow Passed(p, French)$

c) $\exists p1\ Student(p1) \land Took(p1, Greek) \land Spring2001(Greek) \land [\forall p2\ (Student(p2) \land Took(p2, Greek) \land Spring2001(Greek)]]$

d) $Higher(Score(Greek), Score(French))$

e) $\forall p\ [\exists o\ Buy(p, o)] \Rightarrow Smart(p)$

f) $\neg[\exists p, o\ Buy(p, o) \land Expensive(o)]$

g) $\exists p1\ Job(p1, Agent) \land [\forall p2\ Sell(p1, p2) \Rightarrow \neg Insured(p2)]$

h) $\exists p1\ Job(p1, Barber) \land [\forall p2\ (ManInTown(p2) \land \neg Shaves(p2, p2)) \Rightarrow Shaves(p1, p2)]$

i) $\forall p\ (UKBorn(p) \land [(UKResident(Father(p)) \lor UKbyBirth(Father(p)) \lor UKbyDescent(Father(p))) \land (UKResident(Mother(p)) \lor UKbyBirth(Mother(p)) \lor UKbyDescent(Mother(p)))]) \Rightarrow UkbyBirth(p)$

j) $\forall p\ ([UKbyBirth(Father(p)) \lor UKbyBirth(Mother(p))] \land \neg UKBorn(p)) \Rightarrow UKbyDescent(p)$

k) $\forall p1\ Job(p1, Politician) \Rightarrow [(\exists p2\ \forall t\ Fool(p1, p2, t)) \land (\forall p2\ \exists t\ Fool(p1, p2, t)) \land \neg(\forall p2, t\ Fool(p1, p2, t))]$

l) $\forall p1, p2[Greek\_person(p1) \land Greek\_person(p2)] \Rightarrow [\exists l\ Speaks(p1, l) \land Speaks(p2, l)]$