Name: Ajay Reddy tudumula
E_uid: 11580520

Artificial Intelligence - HOMEWORK NO 4

CSCE 5210

Q1. Consider the knowledge base KB:

$a \leftarrow b \wedge c$
$b \leftarrow d$
$b \leftarrow e$
c
$d \leftarrow h$
e
$f \leftarrow g \wedge b$
$g \leftarrow c \wedge k$
$j \leftarrow a \wedge b$.

a) Show how the bottom up proof procedure work for this example. Give all logical sequences of KB.

Ans: The bottom up proof procedure work as follows.

{ }
{ c }
{ c, e }
{ c, e, b }
{ c, e, b, a }
{ c, e, b, a, j }

The Algorithm terminates with $c = \{ c, e, b, a, j \}$

b) f is not a logical sequence of KB. Give a model of KB in which f is false

Ans:- f is not a logical consequence of KB becase when we try to derive above knowledge base bottomup Proof. Then it desives a set {c, e, b, a, } which is all are true. so, the remaining variables {d, f, g, h, k} is not going to be derived so, that's why f is not a logical consequences.

c) a is a logical consequence of KB. Give a top down desivation for the ask a?

Ans:- Considers the starting variable in this top-down model is 'a'. So, the model will desive the following steps.

yes ← a
yes ← b ∧ c
yes ← e ∧ c
yes ← c
yes ←

In this way these sequence of chore will lead to Proof of Solution.

Q2. Consider the following clauses and integrity constraints.

false ← a ∧ b. a ← d b ← d

false ← c a ← g b ← e

c ← h a ← h

Suppose the assumables are {d, e, f, g, h, j} What are the minimal conflicts.

Ans:- From the given close and Integrity constraints there are the minimal conflicts that can be derived

{d, g} → when ((A) d = true & g = true (B)) d ∧ g = true Conflict

{d, h} → when ((A) d = true & h = true (B)) d ∧ h = true Conflict

{e, d} → when ((A) e = true & d = true (B)) end = true Conflict

{e, h} → when ((A) e = true & h = true (B)) e ∧ h = true Conflict

{e, g} → when ((A) e = true & g = true (B)) e ∧ g = true Conflict

{d} → when ((A) d = true & d = true (B)) d ∧ d = true Conflict

{h} → when ((C) h = true) & true Conflict.

These are the minimal Conflict {d, g} {d, h} {e, d} {e, g} {e, h} {d} {h}.

Q.3) Consider the knowledge Base.

1. a ← b ∧ c. 2. b ← d 3) d ← h

4. a ← e ∧ f 5. b ← f ∧ h 6) f ← g

7. c ← e 8. e 9. g ← c

a) Give a model of kb.

A model is a assignment of truth values to the atoms that makes all the sentences in the KB true.

Given that e is true, from (7) c is true.

Q.3) consider the following kb.

$r(a)$ $\quad$ $s(d,b)$

$r(e)$ $\quad$ $s(e,d)$

$p(c)$ $\quad$ $p(X) \leftarrow q(X) \wedge r(X)$

$q(b)$ $\quad$ $q(X) \leftarrow s(X,Y) \wedge q(Y)$.

$s(a,b)$

Show the set of ground atomic consequences derivable from this kb. Use the bottom-up proof procedure assuming, at each iteration, the first applicable clause is selected in the order shown. Furthermore, applicable constant substitutions are chosen in "alphabetic order" if the more than one applies to a given clause for ex X/a and X/b are both applicable. For each clause at some iteration derive $q(a)$. In what order consequence derived.

Ans: When we want to derive atomic consequence then first we have to write all the ground intances. Those are

$q(a)$

$r(e)$

$p(c)$

$s(a,b)$

$s(d,b)$

$s(e,d)$

$f(a) \leftarrow q(a) \wedge r(a)$

$p(b) \leftarrow q(b) \wedge r(b)$

$p(c) \leftarrow q(c) \wedge r(c)$

$p(d) \leftarrow q(d) \wedge r(d)$

$p(e) \leftarrow q(e) \wedge r(e)$

$q(b) \leftarrow s(b,a) \wedge q(a)$

$q(b) \leftarrow s(b,b) \wedge q(b)$

$q(b) \leftarrow s(b,c) \wedge q(c)$

$q(b) \leftarrow s(b,d) \wedge q(d)$

$q(b) \leftarrow s(d,e) \wedge q(e)$

$q(a) \leftarrow s(a,a) \wedge q(a)$

$q(a) \leftarrow s(a,b) \wedge q(b)$

$q(a) \leftarrow s(a,c) \wedge q(c)$

$q(a) \leftarrow s(a,d) \wedge q(d)$

$q(a) \leftarrow s(a,e) \wedge q(e)$

$q(c) \leftarrow s(c,a) \wedge q(a)$

$q(c) \leftarrow s(c,b) \wedge q(b)$

$q(c) \leftarrow s(c,c) \wedge q(c)$

$q(c) \leftarrow s(c,d) \wedge q(d)$

$q(c) \leftarrow s(c,e) \wedge q(e)$

$q(d) \leftarrow s(d,a) \wedge q(a)$

$q(d) \leftarrow s(d,b) \wedge q(b)$

$q(d) \leftarrow s(d,c) \wedge q(c)$

$q(d) \leftarrow s(d,d) \wedge q(d)$

$q(d) \leftarrow s(d,e) \wedge q(e)$

$q(e) \leftarrow s(e,a) \wedge q(a)$

$r(e) \leftarrow s(e,b) \wedge q(b)$

$q(e) \leftarrow s(e,d) \wedge q(c)$

$q(e) \leftarrow s(e,c) \wedge q(c)$

these above are all the ground instances from all these ground instances by using the bottom up procedure we will find the ground atomic consequences.

so, by using bottom up procedure these below are the ground atomic consequence

→ { }

→ { r(a) }

→ { r(a), r(e) }

→ { r(a), r(e), p(c) }

→ { r(a), r(e), p(c), q(b) }

→ { r(a), r(e), p(c), q(b), s(a,b) }

→ { r(a), r(e), p(c), q(b), s(a,b), s(b,d) }

→ { r(a), r(e), p(c), q(b), s(a,b), s(b,d), s(e,d) }

→ { r(a), r(e), p(c), q(b), s(a,b), s(b,d), s(e,d), q(a) }

→ { r(a), r(e), p(c), q(b), s(a,b), s(b,d), s(e,d), q(a), q(d) }

→ { r(a), r(e), p(c), q(b), s(a,b), s(b,d), s(e,d), q(a), q(d), q(e) }

so, the final set of ground atomic consequence that we derived from the bottom up procedure is

{ r(a), r(e), p(c), q(b), s(a,b), s(d,b), s(e,d), q(a) q(d), q(e) }

Q.4) consider the following kb.

has_access (X, library) ← student (X)
has_access (X, library) ← faculty (X).
has_access (X, library) ← has_access (Y, library) ∧ Parent (Y, X
has_access (X, office) ← has_keys (X)
faculty (diane)
student (mary)
Parent (susan, sarah)
Parent (karen, todd)
faculty (ming)
Parent (diane, karen)
Parent (sarah, ariel)
Parent (diane, robyn)
Parent (karen, chelsey)
Student (William

(a) Provide an SLD derivation of query has_access (todd, library)

← has_access (todd, library)
↓ has_access (X, library) ← has_acess (Y, library
Parent (Y, X)
← has_access (Y, library) ∧ Parent (Y, todd)
↓ Parent (karen, todd)
← has_access (karen, library) ∧ Parent (karen, todd

Here we done the two substitions one is X with todd and second one with Y with karen.

So, final set unified set is {todd/X, karen/Y}

b) The query has_access (mary, library) has two SLD derivations. Give both, but do not show the clauses chosen or the substitutions.

Any

First derivation

← has_access (mary_library)

↓ has_access (x, library) ← student (x)

student (mary)

Here we done only one substitution that is X with mary. So final set is {mary/x}

Second derivation

← has_ access (mary, library)

↓ has_ access (x, library) ← has_access (Y, ∧ Parent (Y, x)

← has_access (Y, library) ∧ Parent (Y, mary)

↓ Parent (karen, mary)

Here we done two substitution one is X with mary and other one is Y with karen So the final set is {mary/x, karen/y}

c) Does there exist an SLD derivation for has_access (ariel, library)
Explain Why (or) Why not ?

Ans:- There is an derivation exists for has_access (ariel, library) because there is a supporting constraint that helps is

Parent (sarah, ariel)

So, the derivation will looks like this

← has_access (ariel, library)

↓ has_access (X, library) ← has_access (Y, library)

∧ Parent (Y, X)

← has_access (Y, library) ∧ Parent (Y, ariel)

↓ Parent (sarah, ariel)

← has_access (sarah, library) ∧ Parent (sarah, ariel)

So, here we have done only two substitution one is X with ariel and other one is Y with sarah. So, the final unity set is {ariel/X, sarah/Y}

(d) Explain Why the set of answers to the query has_access (X, office) is empty.

Ans:- Set of answers for the query has_access (X, office) is because of the clause

has_access (X, office) ← has_keys (X)

but for the has_keys (X) there is no derivation thats why the answer for the query is empty.

(e) Suppose the following clause to the kb has_keys(x) ← faculty(
What are the answers to the query has_access(x, Office)?

Ans: If we add the given clause to the kb after
than the SLD derivation for this query has_access
(x, office) that is

← has_access(x, office)

| has_access(x, office) ← has.keys(x)

← has_keys(x)

| has_keys(x) ← faculty(x)

← faculty(x)

The answer set for X is {diane, ming}

Answers to the given query is {diane, ming}