Home work-4 on Logic in AI

naveen ajai kardasu

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

A:- 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

A:- f is not a logical consequences of KB because when we try to derive above knowledge base bottom up proof. Then it derives a set $\{c, e, b, a, j\}$ 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 consequences of KB. Give a top down derivation for the ask a?

A:- consider the starting variable in this top-down model. is 'a' so, the model will deoive the following steps

yes ← a

yes ← b∧c

yes ← e∧c

yes ← c

yes ←

In this way these sequence of choice will lead to proof of solution

Q2:- consider the following clauses

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,i} what are minimal conflicts

A:- from the given close and integrity constraints these 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)) e∧d = 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}

Q3) when we want to derive atomic consequences then first we have to write all the ground instances there are

$\gamma(a)$
$\gamma(e)$
$P(e)$
$\downarrow(b)$
$s(a,b)$
$s(d,b)$
$s(e,d)$
$P(a) \leftarrow \downarrow(a) \wedge \gamma(a)$
$P(b) \leftarrow \downarrow(b) \wedge \gamma(b)$
$P(c) \leftarrow \downarrow(c) \wedge \gamma(c)$
$P(d) \leftarrow \downarrow(d) \wedge \gamma(d)$
$P(e) \leftarrow \downarrow(e) \wedge \gamma(e)$
$\downarrow(b) \leftarrow s(b,a) \wedge \downarrow(a)$
$\downarrow(b) \leftarrow s(b,h) \wedge \downarrow(b)$
$\downarrow(b) \leftarrow s(b,c) \wedge \downarrow(c)$
$\downarrow(b) \leftarrow s(b,d) \wedge \downarrow(d)$
$\downarrow(b) \leftarrow s(b,e) \wedge \downarrow(e)$
$\downarrow(a) \leftarrow s(a,a) \wedge \downarrow(a)$
$\downarrow(a) \leftarrow s(a,b) \wedge \downarrow(b)$
$\downarrow(a) \leftarrow s(a,c) \wedge \downarrow(c)$
$\downarrow(a) \leftarrow s(a,d) \wedge \downarrow(d)$
$\downarrow(a) \leftarrow s(a,e) \wedge \downarrow(e)$
$\downarrow(c) \leftarrow s(c,a) \wedge \downarrow(a)$

$\downarrow(e) \leftarrow s(c,b) \wedge \downarrow(b)$

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

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

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

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

$\downarrow(e) \leftarrow s(e,b) \wedge \downarrow(b)$

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

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

$\downarrow(e) \leftarrow s(e,e) \wedge \downarrow(e)$

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

So by using bottom up procedure these below are the ground atomic consequences

-> { }

-> {r(a)}

-> {r(a), r(e)}

-> {r(a), r(e), r(c)}

-> {r(a), r(e), p(c), $\downarrow$(b)}

-> {r(a), r(e), p(c), $\downarrow$(b)}

-> {r(a), r(e), p(c), $\downarrow$(b), s(a,b)}

-> {r(a), r(e), p(c), $\downarrow$(b), s(a,b), s(d,b)}

-> {r(a), r(e), p(c), $\downarrow$(b), s(a,b), s(d,b), s(e,d)}

-> {r(a), r(e), p(c), $\downarrow$(b), s(a,b), s(d,b), s(e,d), $\downarrow$(a)}

-> {r(a), r(e), p(c), $\downarrow$(b), s(a,b), s(d,b), s(e,d), $\downarrow$(a), $\downarrow$(d)}

-> {r(a), r(e), p(c), $\downarrow$(b), s(a,b), s(d,b), s(e,d), $\downarrow$(a), $\downarrow$(d), $\downarrow$(e)}

so the final of ground atomic consequences that 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)}

Q4:- Provide an SLD derivation of the Query has

a) 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 substitutions one is x with todd & second is y with karen

so final unified set- is {todd/x, karen/y}

Q4

b) The Query has_access (mary, library) has two SLD deviations Give both of them

first derivation :-

← has_access (mary, library)

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

student (mary)

Here we have done only one substitution that is x with mary so final subset is {mary/x}

second derivation :-

← has_access (mary, library)

| has_access (x, library) ← has_access (y, library) ∧ parent (y, x)

← has_access (y, library) ∧ parent (y, mary)

| parent (karen, mary)

← has_access (karen, library) ∧ parent (karen, mary)

Here we done two substitutions one is x with mary & the other is y with karen

So final set is {mary/x, karen/y}

Q4 c) There is deviation for has_access(ariel, library) because there is a supporting constraint that helps is parent(sarah, ariel)

so the deviation 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 two substitution one is x with ariel & other is y with sarah

so, final subset is {ariel/x, sarah/y}

Q4 d) set of answers for the query has_access(x, office) is because of its clause

has_access(x, office) ← has_keys(x)

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

e) if we add the given clauses to knowledge base after that 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}
Answer to the given query is {diane, ming}