

Problema 1

$a \leftarrow b \wedge c.$

$a \leftarrow e \wedge f.$

$b \leftarrow d.$

$b \leftarrow f \wedge h.$

$c \leftarrow e.$

$d \leftarrow h.$

$e.$

$f \leftarrow g.$

$g \leftarrow c.$

1) modello della knowledge base

un modello per la knowledge base data può essere:

Definita funzione modello $A(x)$ dove x è un atomo:

$A(a) = \text{true}$

$A(b) = \text{true}$

$A(c) = \text{true}$

$A(d) = \text{true}$

$A(e) = \text{true}$

$A(f) = \text{true}$

$A(g) = \text{true}$

$A(h) = \text{true}$

2) interpretazione che non sia un modello della knowledge base

un'interpretazione che non sia un

modello della knowledge base può essere

una interpretazione dove una o più proposizioni di suddetta

KB hanno esito falso:

definita funzione interpretazione $B(x)$ dove x è un atomo:

$B(a) = \text{false}$

$B(b) = \text{true}$

$B(c) = \text{true}$

$B(d) = \text{true}$

$B(e) = \text{true}$

$B(f) = \text{true}$

$B(g) = \text{true}$

$B(h) = \text{true}$

3) 2 atomi che siano conseguenza logica della Knowledge Base

applicando l'algoritmo di
bottom-up proof abbiamo trovato che gli
atomi $\{e, c, g, f, a\}$ comportano KB
(gli atomi elencati sono conseguenza logica di KB)

Ad esempio:

$KB \models e$

$KB \models c$

4) 2 atomi che non siano conseguenza logica della KB

Dalla lista di atomi trovata nel punto 3)
possiamo dedurre che:

$KB \not\models b$

$KB \not\models d$

$KB \not\models h$

Problema 2

numeriamo le proposizioni della KB per farne riferimento durante la bottom-up proof.

1. $a \leftarrow b \wedge c.$

2. $b \leftarrow d.$

3. $b \leftarrow e.$

4. $c.$

5. $d \leftarrow h.$

6. $e.$

7. $f \leftarrow g \wedge b.$

8. $g \leftarrow c \wedge k.$

$$9. j \leftarrow a \wedge b.$$

1) Mostrare come la procedura di bottom-up proof funziona per questo esempio. Elencare tutte le conseguenze logiche di KB

{}
 {c} per 4.
 {c,e} per 6.
 {c,e,b} per 3.
 {c,e,b,a} per 1.
 {c,e,b,a,j} per 9.

KB \models c , e , b , a , j

2) f non è una conseguenza logica di KB. Dare un modello nel quale f è falsa

Un modello in cui f è falsa è:

a = false
 b = false
 c = true
 d = false
 e = true
 f = false
 g = false
 h = false
 j = false
 k = false

3) a è una conseguenza logica di KB. Dare una derivazione top-down per la query ask a

yes \leftarrow a
 derivo a in 1.
 yes \leftarrow b \wedge c
 derivo b in 2.
 yes \leftarrow d \wedge c
 derivo d in 5.
 yes \leftarrow h \wedge c

proof fail, h non è in testa ad alcuna produzione

Riprovo derivando b in 3. invece che in 2:

$\text{yes} \leftarrow b \wedge c$

derivo b in 3.

$\text{yes} \leftarrow e \wedge c$

derivo e in 6.

$\text{yes} \leftarrow c$

derivo c in 4.

$\text{yes} \leftarrow$

procedura proof finita

Problema 3

goto_forest \leftarrow *walking*.

get_gun \leftarrow *hunting*.

goto_forest \leftarrow *hunting*.

get_gun \leftarrow *robbing*.

goto_bank \leftarrow *robbing*.

goto_bank \leftarrow *banking*.

fill_withdrawal_form \leftarrow *banking*.

false \leftarrow *banking* \wedge *robbing*.

false \leftarrow *wearing_good_shoes* \wedge *goto_forest*.

assumable walking, hunting, robbing, banking

1) Supponi che *get_gun* è osservato. Quali sono tutte le spiegazioni minimali per questa osservazione?

Dato che *get_gun* è osservato, vediamo in quale produzione è in testa:

get_gun \leftarrow *hunting*.

get_gun \leftarrow *robbing*.

per mantenere *get_gun* a true ci sono 2 modi:

`o hunting o robbing` devono essere assunti ad essere true.

`get_gun` non è specificato da nessuna altra parte, quindi ci sono 2 spiegazioni minimali:

```
{ hunting } e { robbing }.
```

2) Supponi che `get_gun ∧ goto_bank` è osservato. Quali sono tutte le spiegazioni minimali per questa osservazione?

Per mantenere `get_gun` a true si applica lo stesso procedimento sopra, con spiegazioni al momento `{ hunting }` e `{ robbing }`

in quanto a `goto_bank` abbiamo le 2 produzioni:

```
goto_bank ← robbing.
```

```
goto_bank ← banking.
```

visto che `robbing` è il corpo anche di una proposizione con in testa `get_gun` una spiegazione minimale rimane al momento `{ robbing }`

nell'altro caso abbiamo `banking`, quindi la spiegazione minimale `{ hunting }` la uniamo con `banking`
`= { hunting , banking }`

`{ robbing , banking }` **non** può essere una spiegazione poiché la proposizione:

```
false ← banking ∧ robbing
```

impedisce che entrambi gli atomi siano assumibili.

In conclusione le spiegazioni minimali per l'osservazione `get_gun ∧ goto_bank` sono 2:

```
{ robbing } and { hunting , banking }
```

3) C'è qualcosa che può essere osservato per rimuovere una di queste come una spiegazione minimale? Cosa deve essere aggiunto per rendere possibile spiegarlo?

per rimuovere una spiegazione minimale si può rendere osservabile `fill_withdrawal_form`, tale atomo è in testa a `banking`, quindi in ogni spiegazione minima `banking` deve essere true, questo genera un conflitto con `robbing` per la proposizione:

```
false ← banking ∧ robbing
```

quindi `{ robbing }` non può più essere una spiegazione minimale, rimarrebbe:

```
{ hunting , banking }
```

aggiungendo

```
fill_withdrawal_form ← robbing
```

renderebbe { robbing } spiegabile minimalmente, in quanto sarebbe nel corpo di tutti e tre degli atomi osservati (get_gun , goto_bank , fill_withdrawal_form)

4) Supponi che goto_bank è osservato. Quali sono tutte le spiegazioni minimali per questa osservazione?

Le spiegazioni minimali per goto_bank sono :

```
{ robbing }, { banking }
```

5) Supponi che goto_bank \wedge get_gun \wedge fill_withdrawal_form è osservato. Quali sono tutte le spiegazioni minimali per questa osservazione?

La spiegazione minimale per goto_bank \wedge get_gun \wedge fill_withdrawal_form è una:

```
{ hunting , banking }
```

Problema 4

send_signal_lg_sc \leftarrow *ok_sc_lg* \wedge *alive_sc*.

send_signal_hg_sc \leftarrow *ok_sc_hg* \wedge *alive_sc*.

get_signal_s1 \leftarrow *send_signal_hg_sc* \wedge *ok_s1_ant*.

get_signal_s2 \leftarrow *send_signal_hg_sc* \wedge *ok_s2_ant*.

send_signal_s1 \leftarrow *get_signal_s1* \wedge *ok_s1_trans*.

send_signal_s2 \leftarrow *get_signal_s2* \wedge *ok_s2_trans*.

get_signal_gc \leftarrow *send_signal_s1* \wedge *ok_a1*.

get_signal_gc \leftarrow *send_signal_s2* \wedge *ok_a2*.

get_signal_gc \leftarrow *send_signal_lg_sc* \wedge *ok_a3* \wedge *no_dist*.

- observed: *ok_a1* , *ok_a2* , *ok_a3* , *ok_s1_trans* , *no_signal_gc*
- unsure about: *alive_sc* , *ok_sc_lg* , *ok_sc_hg* , *ok_s*_ant* , *ok_s2_trans* , *no_dist*

1) Specifica un insieme di assumables e un constraint di integrità che modellino la situazione

insieme di assumibili $A = \{ ok_sc_hg, alive_sc, ok_s1_ant, ok_s2_trans, ok_s2_ant, ok_sc_hg, no_dist \}$

il constraint da aggiungere è:

$false \leftarrow get_signal_gc \wedge no_signal_gc.$

2) Usando gli assumables e gli integrity constraints dal punto 1, qual'è l'insieme dei conflitti minimi?

l'insieme dei conflitti minimi è:

$C = \{ \{ ok_sc_hg, alive_sc, ok_s1_ant \}, \{ ok_s2_trans, ok_sc_hg, alive_sc, ok_s2_ant \}, \{ ok_sc_lg, alive_sc, no_dist \} \}$

3) Qual è la diagnosi basata su consistency per la situazione data? in altre parole, quali sono le possibili combinazioni di assunzioni violate che potrebbero motivare il perché il centro di controllo non può ricevere segnale dalla navicella?

la diagnosi basata sulla consistenza è la seguente:

tutti e 3 i punti sono veri:

- 1 o 2 tra le assunzioni $\{ ok_sc_hg, alive_sc, > ok_s1_ant \}$ sono violate.
- da 1 a 3 delle assunzioni $\{ ok_s2_trans, ok_sc_hg, > alive_sc, ok_s2_ant \}$ sono violate.
- 1 o 2 tra le assunzioni $\{ ok_sc_lg, alive_sc, > no_dist \}$ sono violate.

4) Spiega perché la NASA vorrebbe usare l'abduction invece che diagnosi basate sulla consistenza per il dominio

Alla NASA conviene usare l'abduzione perché rende più chiara e immediata da ottenere la lista di componenti che potrebbero aver fallito, e avere in aggiunta delle ipotesi per quelli che funzionerebbero ancora. In quanto per essa sarebbe necessario assiomatizzare il dominio in modo da rendere possibili explanations dove viene segnalato il malfunzionamento di un componente insieme al funzionamento di un altro, in tal modo sarebbe più chiaro il problema. La diagnosi invece si limita a fornire quali componenti violano la consistenza del domain.

5) Si supponga che un disturbo atmosferico $dist$ possa produrre un segnale statico o assente nel segnale a bassa larghezza di banda. Per ricevere il segnale statico, l'antenna $a3$ e il trasmettitore a bassa larghezza di banda del veicolo spaziale sc_lg devono funzionare. Se $a3$ o sc_lg non funzionano o sc è dead, non c'è segnale.

Quali regole e assumibili devono essere aggiunte alla base di conoscenza in modo da poter spiegare le possibili osservazioni no_signal_gc , get_signal_gc o $static_gc$? Si possono ignorare i collegamenti ad alta larghezza di banda. Si possono inventare tutti i simboli necessari

- $dist \rightarrow$ usato per il disturbo atmosferico, viene utilizzato scrivendo $\neg no_dist$

- `no_signal_gc` -> nuovo
- `get_signal_gc` -> già presente nelle produzioni
- `static_gc` -> nuovo

le regole da aggiugnere sono:

```
static_gc ← ok_a3 ∧ ok_sc_lg ∧ ¬no_dist
no_signal_gc ← ¬ok_a3 ∨ ¬ok_sc_lg ∨ ¬alive_sc
false ← static_gc ∧ no_signal_gc ∧ get_signal_sc
false ← static_gc ∧ no_dist
false ← get_signal_sc ∧ ¬no_dist
```

Assumables: `alive_sc` , `ok_sc_lg` , `ok_a3` , `no_dist`