Grundlagen der Wissensverarbeitung – Tutorial 7, Gruppe 4

Arne Beer MN 6489196 Marta Nevermann MN 6419716
Daniel Waller MN 6813853 Julius Hansen MN 6455291

Exercise 1.2

Zuerst werden aus der den Aussagen der Aufgabe Atome (atoms) entnommen:

- gärtner_mordete
- butler_mordete
- gärtner_arbeitete
- butler_arbeitete
- gärtner_schmutzige_hände
- butler_schmutzige_hände
- gärtner_saubere_hände
- butler_saubere_hände
- gärtner_lügt
- butler_lügt

Anschließend können anhand der Aussagen Regeln (rules) erstellt werden:

- butler_arbeitete $\leftarrow$ butler_schmutzige_hände
- gärtner_arbeitete $\leftarrow$ gärtner_schmutzige_hände
- $\neg$butler_arbeitete $\leftrightarrow$ butler_saubere_hände
- $\neg$gärtner_arbeitete $\leftrightarrow$ gärtner_saubere_hände
- $\neg$butler_mordete $\leftarrow$ butler_arbeitete
- $\neg$gärtner_mordete $\leftarrow$ gärtner_arbeitete

Es ist außerdem die Vorraussetzung (integrity constraints) gegeben, dass folgendes gilt:

- false $\leftarrow$ butler_schmutzige_hände $\wedge$ butler_saubere_hände
- false $\leftarrow$ gärtner_schmutzige_hände $\wedge$ gärtner_saubere_hände

Die Aussagen der Verdächtigen stellen hier unsere Assumables dar:

- gärtner_arbeitete
- butler_arbeitete

Mit den Observables:

- gärtner_saubere_hände
- butler_schmutzige_hände

Mit der Annahme, dass beide den Mord nicht begangen haben, was sich aus den Assumables schließen lässt, kann nun ein mimimaler Konflikt erstellt werden:

{¬gärtner_arbeitete}

Daraus folgt: KB $\models$ ¬gärtner_arbeitete

Exercise 1.3

Generelle Annahmen aus mangelndem Know-how:

- Die Fuel pump arbeitet auch wenn der Tank kein Benzin enthält und/oder der Filter kaputt ist
- Der Starter ist nur zu hören wenn die Batterie Ladung hat
- Die Fuel pump arbeitet nicht wenn die electronic fuel regulation beschädigt ist
- Der Motor ist nur zu hören wenn starter und fuel pump auch gehen

Kürzel

- battery : bat
- ignition key : ik
- electronic fuel regulation : efr
- starter : st
- engine : en
- filter : fi
- fuel pump : fp
- fuel tank : ft

$Assumables = \{ok\_st, ok\_fp, ok\_en, turned\_ik, live\_bat, ok\_efr, filled\_ft, ok\_fi\}$

KB = { false ← noise_1 ∧ silent_1.
false ← noise_2 ∧ silent_2.
false ← noise_3 ∧ silent_3.
noise1 ← ok_st ∧ hasPower.
noise_2 ← ok_fp ∧ hasPower ∧ ok_efr.
noise_3 ← ok_en ∧ ok_st ∧ ok_fp ∧ hasPower ∧ getsFuel.
hasPower ← turned_ik ∧ live_bat.
getsFuel ← filled_ft ∧ ok_fi. }

case: no noise observed

silent_1, silent_2, silent_3 = true
Initial state: q = false, ac = yes ← false ∧ false ∧ false

Iter step 1:

- select 'false'
- choose clause 'false ← noise_1 ∧ silent_1.' from KB
- replace false with body of clause

State 1: q = false, ac = yes ← noise_1 ∧ silent_1 ∧ false ∧ false

Iter step 2:

- select 'noise_1'
- choose clause 'noise1 ← ok_st ∧ hasPower.' from KB
- replace 'noise_1' with body of clause

State 2: q = false, ac = yes← ok_st ∧ hasPower ∧ silent_1 ∧ false ∧ false

Iter step 3:

- select 'hasPower' (ok_st is assumabledon't select it)
- choose clause 'hasPower ← turned_ik ∧ live_bat.' from KB
- replace 'hasPower' with body of clause

State 3: q = false, ac = yes ← ok_st ∧ turned_ik ∧ live_bat ∧ silent_1 ∧ false ∧ false

Iter step 4:

- select 'false' (all others are assumable)
- choose clause 'false ← noise_2 ∧ silent_2.' from KB
- replace 'false' with body of clause

State 4: q = false, ac = yes ← ok_st ∧ turned_ik ∧ live_bat ∧ silent_1 ∧ noise_2 ∧ silent_2 ∧ false

Iter step 5:

- select 'noise_2'
- choose clause 'noise_2 ← ok_fp ∧ hasPower ∧ ok_efr.' from KB
- replace 'noise_2' with body of clause

State 5: q = false, ac = yes ← ok_st ∧ turned_ik ∧ live_bat ∧ silent_1 ∧ ok_fp ∧ hasPower ∧ ok_efr ∧ silent_2 ∧ false

Iter step 6:

- select 'hasPower'
- choose clause 'hasPower ← turned_ik ∧ live_bat.' from KB
- replace 'hasPower' with body of clause

State 6: q = false, ac = yes ← ok_st ∧ turned_ik ∧ live_bat ∧ silent_1 ∧ ok_fp ∧ turned_ik ∧ live_bat ∧ ok_efr ∧ silent_2 ∧ false

Iter step 7:

- select 'false'
- choose clause 'false ← noise_3 ∧ silent_3.' from KB
- replace 'false' with body of clause

State 7: q = false, ac = ok_st ∧ turned_ik ∧ live_bat ∧ silent_1 ∧ ok_fp ∧ turned_ik ∧ live_bat ∧ ok_efr ∧ silent_2 ∧ noise_3 ∧ silent_3

Iter step 8:

- select 'noise_3'

- choose clause ‘noise_3 ← ok_en ∧ ok_st ∧ ok_fp ∧ hasPower ∧ getsFuel’ from KB
- replace ‘noise_3’ with body of clause

State 8: q = false, ac = yes ← ok_st ∧ turned_ik ∧ live_bat ∧ silent_1 ∧ ok_fp ∧ turned_ik ∧ live_bat ∧ ok_efr ∧ silent_2 ∧ ok_en ∧ ok_st ∧ ok_fp ∧ hasPower ∧ getsFuel ∧ silent_3

Iter step 9:

- select ‘hasPower’
- choose clause ‘hasPower ← turned_ik ∧ live_bat.’ from KB
- replace ‘hasPower’ with body of clause

State 9: q = false, ac = yes ← ok_st ∧ turned_ik ∧ live_bat ∧ silent_1 ∧ ok_fp ∧ turned_ik ∧ live_bat ∧ ok_efr ∧ silent_2 ∧ ok_en ∧ ok_st ∧ ok_fp ∧ turned_ik ∧ live_bat ∧ getsFuel ∧ silent_3

Iter step 10:

- select ‘getsFuel’
- choose clause ‘getsFuel ← filled_ft ∧ ok_fi.’ from KB
- replace ‘getsFuel’ with body of clause

State 10: q = false ac = yes ← ok_st ∧ turned_ik ∧ live_bat ∧ silent_1 ∧ ok_fp ∧ turned_ik ∧ live_bat ∧ ok_efr ∧ silent_2 ∧ ok_en ∧ ok_st ∧ ok_fp ∧ turned_ik ∧ live_bat ∧ filled_ft ∧ ok_fi ∧ silent_3

Zusammengefasst:
yes ← ok_st ∧ turned_ik ∧ live_bat ∧ silent_1 ∧ ok_fp ∧ ok_efr ∧ silent_2 ∧ ok_en ∧ filled_ft ∧ ok_fi ∧ silent_3

Minimal conflicts : sehr viele. Entweder jedes Teil individuell kann kaputt sein oder aber Teile in beliebiger Kombination, solange sie keine Teilmengen von bereits aufgeführten Konflikten sind.

case: Only noise 1

Algorithmus analog zu oben...

Minimal conflicts: {ok_en, ok_fp}, {ok_efr} Entweder kann die electronic fuel regulation kaputt sein und somit die nachfolgende Kette nicht funktionieren oder engine und fuel pump sind kaputt oder beliebige Kombinationen der 3 Elemente sind kaputt

case: Only noise 2

Algorithmus analog zu oben...

Minimal conflicts: {ok_en, ok_st}, {ok_st, ok_fp, filled_ft}, {ok_st, ok_fp, ok_fi} Entweder sind der Starter und der Motor defekt oder der Starter in Kombination mit einem oder beiden Teilen aus Filter und Fuel Tank

case: noise 1&2 but not noise 3

Algorithmus analog zu oben...

Minimal conflicts: {ok_en}, {ok_fi}, {ok_ft} Entweder eines oder jede Kombination dieser Teile könnte kaputt sein