Contents

1 Language

1.1 Integrity constraint

An integrity constraint of the form:

$$\leftarrow a_1, \ldots, a_m, \neg a_{m+1}, \ldots, \neg a_n$$

can be written as a normal rule:

$$x \leftarrow a_1, \ldots, a_m, \neg a_{m+1}, \ldots, \neg a_n, \neg x$$

1.2 Choice rules

A choice rule of the form:

$$\{a_1, \ldots, a_m\} \leftarrow a_{m+1}, \ldots, a_n, \neg a_{n+1}, \ldots, \neg a_o$$

can be written as $2m + 1$ normal rules:

$$\begin{array}{rcl} x & \leftarrow & a_{m+1}, \ldots, a_n, \neg a_{n+1}, \ldots, \neg a_o \\ x_1 & \leftarrow & \neg a_1 \\ \vdots & & \vdots \\ x_m & \leftarrow & \neg a_m \\ a_1 & \leftarrow & x, \neg x_1 \\ \vdots & & \vdots \\ a_m & \leftarrow & x, \neg x_m \end{array}$$

1.3 Cardinality rules

1.3.1 CR with lower bound

A cardinality rule of the from:

$$a_0 \leftarrow l\{a_1, \ldots, a_m, \neg a_{m+1}, \ldots, \neg a_n\}$$

can be translated into the normal rules:

$$\begin{array}{rcl} a_0 & \leftarrow & x(1, l) \\ x(i, k+1) & \leftarrow & x(i+1, k), a_i \\ x(i, k) & \leftarrow & x(i+1, k) \\ x(j, k+1) & \leftarrow & x(j+1, k), \neg a_j \\ x(j, k) & \leftarrow & x(j+1, k) \end{array}$$

with:
$0 \leqslant k \leqslant l$,
$1 \leqslant i \leqslant m$
$m + 1 \leqslant j \leqslant n$

1.3.2 CR with lower and upper bound

A cardinality rule of the from:

$$a_0 \leftarrow l\{a_1, \ldots, a_m, \neg a_{m+1}, \ldots, \neg a_n\}u$$

can be translated into normal rules:

$$\begin{array}{rcl} a_0 & \leftarrow & x, \neg y \\ x & \leftarrow & l\{a_1, \ldots, a_m, \neg a_{m+1}, \ldots, \neg a_n\} \\ y & \leftarrow & u + 1\{a_1, \ldots, a_m, \neg a_{m+1}, \ldots, \neg a_n\} \end{array}$$

1.3.3 CR in the head

A cardinality rule of the from:

$$l\{a_1, \ldots, a_m, \neg a_{m+1}, \ldots, \neg a_n\}u \leftarrow a_{n+1}, \ldots, a_o, \neg a_{o+1}, \ldots, \neg a_p$$

can be translated into the normal rules:

$$\begin{array}{rcl} x & \leftarrow & a_{n+1}, \ldots, a_o, \neg a_{o+1}, \ldots, \neg a_p \\ \{a_1, \ldots, a_m\} & \leftarrow & x \\ y & \leftarrow & l\{a_1, \ldots, a_m, \neg a_{m+1}, \ldots, \neg a_n\}u \\ & \leftarrow & x, \neg y \end{array}$$

1.3.4 Full fledged CR

A rule of the form:

$$l_0\ S_0\ u_0 \leftarrow l_1\ S_1\ u_1, \ldots, l_n\ S_n\ u_n$$

can be translated into normal rules:

$$\begin{array}{rcl} x & \leftarrow & y_1, \ldots, y_n, \neg z_1, \ldots, \neg z_n \\ S_0^+ & \leftarrow & x \\ & \leftarrow & x, \neg y_0 \\ & \leftarrow & x, z_0 \\ y_i & \leftarrow & l_i\ S_i \\ z_i & \leftarrow & u_i + 1\ S_i \end{array}$$

2 Program definition

2.1 Herbrand universe $\mathcal{T}$ & base $\mathcal{A}$

2.1.1 Definition

- $\mathcal{T}$: set of variable-free terms
- $\mathcal{A}$: set of variable-free atoms constructed from $\mathcal{T}$

$$ground(r) = \{r\theta \mid \theta : var(r) \rightarrow \mathcal{T} \text{ and } var(r\theta) = \emptyset\} \tag{1}$$

$$ground(P) = \bigcup_{r \in P} ground(r) \tag{2}$$

A set X of ground atoms is a stable model of P, if X is a stable model of $ground(P)$

2.1.2 Fill out

$$\mathcal{T} = \{atom1, atom2, atom3, ..\}$$

$$\mathcal{A} = \{atom1, atom2, atom3, ..\}$$

$$ground(P) = \left\{ \begin{array}{lcl} head & \leftarrow & body \\ head & \leftarrow & body \\ head & \leftarrow & body \\ head & \leftarrow & body \end{array} \right\}$$

2.2 Fill out

$$P = \left\{ \begin{array}{lcl} head & \leftarrow & body \\ head & \leftarrow & body \\ head & \leftarrow & body \\ head & \leftarrow & body \end{array} \right\}$$

3 Consequences $Cn(P)$ of P

3.1 Definition

The consequences of a program P $Cn(P)$ is the smallest set **closed** under P. A set X of atoms is closed under a positive program P if $h(r) \in X$ whenever $B(r)^+ \subseteq X$ for all $r \in P$.

3.2 Description

- what can be derived from the positive logic program P using its rules and facts

3.3 Fill out

$$Cn(P) = \{atom1, atom2, atom3, ..\}$$

4 $\mathrm{T_P}$-Operator

4.1 Definition

Iteratively calculating the consequences of a program $Cn(P)$, by applying:

$$T_P X = \{h(r) \mid r \in P,\ B(r)^+ \subseteq X,\ B(r)^- \cap Y = \emptyset\} \tag{3}$$

4.2 Description

- initialize $T_P^0(X)$
- apply iteration: $T_P^i(X) = T_P(T_P^{i-1}(X))$
- stop if $T_P(X_i) = T_P(X_{i+1})$
- $Cn(P) = \bigcup_{i \geqslant 0} T_P^i(\varnothing)$

4.3 Fill out

	X	$T_P(X)$
X_0	$\varnothing$	$\{atom1, atom2, atom3\}$
X_1	$\{atom1, atom2, atom3\}$	$\{atom1, atom2, atom3\}$
X_2	$\{atom1, atom2, atom3\}$	$\{atom1, atom2, atom3\}$
X_3	$\{atom1, atom2, atom3\}$	$\{atom1, atom2, atom3\}$

5 Reduct $\mathbf{P^X}$

5.1 Definition

The reduct P^X of a program P relative to a set X of atoms is defined as:

$$P^X = \{h(r) \leftarrow B(r)^+ \mid r \in P, B(r)^- \cap X = \varnothing\} \tag{4}$$

5.2 Description

- get rid of rules satisfying $B(r)^- \cap X = \varnothing$
- get rid of remaining negative body literals
- a set X is a stable model of P if:
  - $Cn(P^X) = X$

5.3 Fill out

$$P^X = \left\{ \begin{array}{lcl} head & \leftarrow & body \\ head & \leftarrow & body \\ head & \leftarrow & body \\ head & \leftarrow & body \end{array} \right\}$$

6 Simple Solving algorithm

6.1 Definition

Approximate a stable model X by L a lower bound X and by U an upper bound on X. Use the expand procedure:

$$expand_P = \left\{ \begin{array}{lcl} L' & \leftarrow & L \\ U' & \leftarrow & U \\ L & \leftarrow & L' \cup Cn(P^{U'}) \\ U & \leftarrow & U' \cap Cn(P^{L'}) \end{array} \right\}$$

- at each step L becomes larger or stays in size
- at each step U becomes smaller or stays in size
- continue expand procedure until $L = L'$ and $U = U'$
- $L = U \Rightarrow L$ is a stable model of P
- $L \nsubseteq U \Rightarrow P$ has no stable model

6.2 Description

1. start with $L' = \emptyset$ and $U' = A(P)$
2. compute the consequences of L' and $U' \Rightarrow Cn(L')$, $Cn(U')$
3. for L compute the union $L' \cup Cn(P^{U'})$
4. for U compute the intersetction $U' \cap Cn(P^{L'})$
5. if $L = L'$ and $U = U'$ is not reachible make a guess:
   (a) set one atom a to true
   (b) start with $L' = a$ and $U' = A(P)$
   (c) continue from step 2.
6. if $L = L'$ and $U = U'$ is reached
   (a) compute also the complement of the guess $\neg a$
   (b) start with $L' = \emptyset$ and $U' = A(P) \setminus a$
   (c) continue from step 2.

6.3 Fill out

L'	$Cn(P^{U'}$	L	U'	$Cn(P^{L'}$	U
$\emptyset$	$\{atom1, atom2, atom3\}$	$\{a1, a2, a3\}$	$\{a1, a2, a3\}$	$\{a1, a2, a3\}$	$\{a1, a2, a3\}$
$\{a1, a2, a3\}$	$\{a1, a2, a3\}$	$\{a1, a2, a3\}$	$\{a1, a2, a3\}$	$\{a1, a2, a3\}$	$\{a1, a2, a3\}$
$\{a1, a2, a3\}$	$\{a1, a2, a3\}$	$\{a1, a2, a3\}$	$\{a1, a2, a3\}$	$\{a1, a2, a3\}$	$\{a1, a2, a3\}$
$\{a1, a2, a3\}$	$\{a1, a2, a3\}$	$\{a1, a2, a3\}$	$\{a1, a2, a3\}$	$\{a1, a2, a3\}$	$\{a1, a2, a3\}$

7 Clarks Completion $CF(P)$

7.1 Definition

Let P be a normal logic program. Clarks Completion $CF(P)$ is defined as:

$$CF(P) = \left\{a \leftrightarrow \bigvee_{r \in P,\, h(r)=a} BF(B(r)) \mid a \in A(P)\right\} \tag{5}$$

with $BF(B(r))$:

$$BF(B(r)) = \bigwedge_{a \in B(r)^+} a \;\wedge \bigwedge_{a \in B(r)^-} \neg a \tag{6}$$

7.2 Description

- find rules with the same head
- combine the body literals of a rule with an and-operator $\wedge$ (6)
- combine all rules with an or-operator $\vee$ (5)

Every stable model of P is a model of $CF(P)$, but not vice versa. The models of $CF(P)$ are called **supported models**.

7.3 Usefull symbols

- $\wedge$
- $\vee$
- $\neg$
- $\leftrightarrow$
- $\rightarrow$
- $\leftarrow$
- $\perp$
- $\top$

7.4 Fill out

$$CF(P) = \left\{ \begin{array}{ccc} a & \leftrightarrow & (\neg b \wedge c) \vee d \\ a & \leftrightarrow & (\neg b \wedge c) \vee d \\ a & \leftrightarrow & (\neg b \wedge c) \vee d \end{array} \right\}$$

8 Positive atom dependency graph

8.1 Definition

$$G(P) \;=\; (A(P),\; \{(b, a) \mid r \in P,\; b \in B(r)^+,\; h(r) = a\})$$

8.2 Description

- go through each rule of $CF(P)$ or P
- for each positive body literal b in a rule create a tuple $(b,\ a)$, where a is the head of the rule
- If $G(P)$ is acyclic $\rightarrow$ P is called **tight**
- $\Rightarrow$ supported models = stable models
- build graph and get tex code from: `http://madebyevan.com/fsm/`

8.3 Fill out

$$\begin{aligned} G(P) = (\{a, b, c, d, ...\}, \\ \{(b, a), (c, a), (d, a), \\ (b, b), (c, b), (d, b), \\ (b, c), (c, c), (d, c)\}) \end{aligned}$$

9 Loop

9.1 Definition

Let $G(P) = (A(P), E)$ be the positive atom dependency graph of a normal logic program P.
A set $\emptyset \subset L \subseteq A(P)$ is a **loop** of P, if it induces a non-trivial strongly connected subgraph of $G(P)$: $\Rightarrow$ *loop*(P).
loop$(P) = \emptyset \Rightarrow P$ is acyclic/**tight**.

9.2 Description

- draw the graph
- look for $\longleftrightarrow$
- look for cycles

Note: If there are 2 loops with a common node the union of nodes is a loop as well.

9.3 Fill out

$$\begin{aligned} loop(P) = \{\{loop1\}, \\ \{loop2\}, \\ \{loop3\}\} \end{aligned}$$

10 Loop Formula

10.1 Definition

Let P be a normal logic program for $L \subseteq A(P)$, define the **external support** of L for P as:

$$ES_P(L) = \{r \in P \mid h(r) \in L,\ B(r)^+ \cap L = \varnothing\} \tag{7}$$

It follows the **Loop Formula** for a loop $L \in loop(P)$:

$$LF_P(L) = (\textstyle\bigvee_{a \in L} a) \rightarrow \left(\textstyle\bigvee_{r \in ES_P(L)} BF(B(r))\right) \tag{8}$$

The **Loop Formula** for P is:

$$LF_P(L) = \{LF(L) \mid L \in loop(P)\} \tag{9}$$

10.2 Description

- Compute the external support $ES_P(L)$ for each loop:
  - take the rule of P if:
    1. the head of the rule is in L
    2. the positive body literal is not in L
- combine the atoms of the loop L with or-oprator $\vee$
- $\rightarrow$
- combine the rules of the **external support** with an or-oprator $\vee$

$\Rightarrow$ if a supported model is also a model of $LF(P)$ it is a stable model.

10.3 Usefull symbols

- $\wedge$
- $\vee$
- $\neg$
- $\rightarrow$
- $\leftarrow$

10.4 Fill out $ES_P(L)$

$$ES_P(L) = \left\{ \begin{array}{ccc} head & \leftarrow & body \\ head & \leftarrow & body \\ head & \leftarrow & body \\ head & \leftarrow & body \end{array} \right\}$$

10.5 Fill out $LF(P)$

$$LF(P) = \left\{ \begin{array}{lcl} loop1 & \rightarrow & ES_P(loop1)rule1 \vee ES_P(loop1)rule2 \\ loop2 & \rightarrow & ES_P(loop2)rule1 \vee ES_P(loop1)rule2 \\ loop3 & \rightarrow & ES_P(loop3)rule1 \vee ES_P(loop1)rule2 \end{array} \right\}$$

11 Unfounded Sets

11.1 Definition

Let P be a normal logic program and let $\langle T, F \rangle$ be a partial interpretation. A set $U \subseteq A(P)$ is an unfounded set of P with respect to $\langle T, F \rangle$ if for each rule $r \in P$ such that $h(r) \in U$ one of the condition holds:

1. $B(r)^+ \cap F \neq \emptyset$
2. $B(r)^- \cap T \neq \emptyset$
3. $B(r)^+ \cap U \neq \emptyset$

11.2 Description

- $\emptyset$ is an unfounded set per definition
- find the greatest unfounded set U_P with respect to $\langle T, F \rangle$ (compare: 13.2)
- check all subsets of U_P $\langle T, F \rangle$ for the conditions of a unfounded set

11.3 Usefull symbols

- $\wedge$
- $\vee$
- $\neg$
- $\rightarrow$
- $\rangle$
- $\langle$
- $\perp$
- $\top$

11.4 Fill out all unfounded sets $u_p \langle T, F \rangle$ of P

$$\begin{gathered} u_p \langle T, F \rangle = \{\{\emptyset\}, \\ \{unfoundedset1\}, \\ \{unfoundedset2\}, \\ \{unfoundedset3\}\} \end{gathered}$$

Note: I used $u_p \langle T, F \rangle$ (lower case u) as the set of unfounded sets.

11.5 Fill out loop formulas for the unfounded sets $u_p \left\langle T, F \right\rangle$ of P

$$LF(unfoundedset) = \left\{ \; atomsofunfoundedset \;\; \rightarrow \;\; ES_P(unfoundedset) \; \right\}$$

11.6 Fill out table of unfounded sets and loop formulas

U	$LF_P(U)$
$\emptyset$	$\bot \rightarrow \bot$
$U1$	$a1 \vee a2 \vee a3 \rightarrow ES_P(U1)$
$U2$	$a1 \vee a2 \vee a3 \rightarrow ES_P(U2)$
$U3$	$a1 \vee a2 \vee a3 \rightarrow ES_P(U3)$

11.7 Fill out greatest unfounded set $U_P \left\langle T, F \right\rangle$ of P

$$U_p \left\langle T, F \right\rangle = \{greatest\ unfounded\ set\}$$

12 Fitting semantics

12.1 Definition

Let P be a normal logic program and $\langle T, F \rangle$ a partial interpretation. The **fitting operator** Φ_P is defined as:

$$\Phi_P\langle T, F\rangle = \langle T_P\langle T, F\rangle, F_P\langle T, F\rangle\rangle \tag{10}$$

$$T_P\langle T, F\rangle = \{h(r) \mid r \in P, B(r)^+ \subseteq T,\ B(r)^- \subseteq F\} \tag{11}$$

$$F_P\langle T, F\rangle = \{a \in A(P) \mid B(r)^+ \cap F \neq \emptyset \text{ or } B(r)^- \cap T \neq \emptyset,\ \forall r \in P \text{ s.t. } h(r) = a\} \tag{12}$$

Using Φ_P iteratively leads to the **fitting semantics**:

$$\begin{aligned} \Phi_P^0\langle T, F\rangle &= \langle T, F\rangle \\ \Phi_P^{i+1}\langle T, F\rangle &= \Phi_P \Phi_P^i\langle T, F\rangle \\ &\Rightarrow \textstyle\bigsqcup_{i \geqslant 0} \Phi_P^i\langle \emptyset, \emptyset\rangle \end{aligned}$$

- $\Phi_P\langle T, F\rangle = \langle T, F\rangle$ is called fitting fixpoint
- the fitting semanstics corresponds with the **least fixpoint**
- the least fixpoint can be expanded
- the **total fixpoints** corresponds to the supported models

12.2 Description

- for each iteration compute T_P and F_P:
- T_P:
    1. search in P for positive body literals, which are also in T of the current partial interpretation

2. search in P for negative body literals, which are also in F of the current partial interpretation
3. if a rule is satisfied add its head to T

- F_P:
  1. search in P for positive body literals, which are in F of the current partial interpretation $\Rightarrow$ rule becomes inapplicable
  2. search in P for negative body literals, which are in T of the current partial interpretation $\Rightarrow$ rule becomes inapplicable
  3. if all rules for a paticular head are inapplicable $\Rightarrow$ add this atom to F
- repeat until $\Phi_P^i\langle T, F\rangle = \Phi_P^{i+1}\langle T, F\rangle$

12.3 Fill out

$$\begin{aligned}
\Phi_P^0\langle\emptyset,\emptyset\rangle &= \langle\emptyset,\emptyset\rangle \\
\Phi_P^1\langle\emptyset,\emptyset\rangle &= \langle\{a1,a2\},\{a1,a2\}\rangle \\
\Phi_P^2\langle\emptyset,\emptyset\rangle &= \langle\{a1,a2\},\{a1,a2\}\rangle \\
\Phi_P^3\langle\emptyset,\emptyset\rangle &= \langle\{a1,a2\},\{a1,a2\}\rangle \\
\Phi_P^4\langle\emptyset,\emptyset\rangle &= \langle\{a1,a2\},\{a1,a2\}\rangle \\
\textstyle\bigsqcup_{i\geqslant 0}\Phi_P^i\langle\emptyset,\emptyset\rangle &= \langle\{a1,a2\},\{a1,a2\}\rangle
\end{aligned}$$

13 Well-founded semantics

13.1 Definition

Let P be a normal logic program and $\langle T, F\rangle$ a partial interpretation. The **well-founded operator** Ω_P is defined as:

$$\Omega_P\langle T, F\rangle = \langle T_P\langle T, F\rangle, U_P\langle T, F\rangle\rangle \tag{13}$$

Recall: U_P is the **Greatest Unfounded Set**.

$$T_P\langle T, F\rangle = \{h(r) \mid r \in P, B(r)^+ \subseteq T,\ B(r)^- \subseteq F\} \tag{14}$$

$$U_P\langle T, F\rangle = A(P)\backslash Cn(\{r \in P \mid B(r)^+ \cap F = \emptyset\}^T) \tag{15}$$

Note:

$$\begin{aligned}
&Cn(\{r \in P \mid B(r)^+ \cap F = \emptyset\}^T) = \\
Cn(\{h(r) \leftarrow B(r)^+ \mid r \in P, &B(r)^+ \cap F = \emptyset,\ B(r)^- \cap T = \emptyset\})
\end{aligned}$$

Using Ω_P iteratively leads to the **well-founded semantics**:

$$\begin{aligned}
\Omega_P^0\langle T, F\rangle &= \langle T, F\rangle \\
\Omega_P^{i+1}\langle T, F\rangle &= \Omega_P\Omega_P^i\langle T, F\rangle \\
&\Rightarrow \textstyle\bigsqcup_{i\geqslant 0}\Omega_P^i\langle\emptyset,\emptyset\rangle
\end{aligned}$$

13.2 Description

- for each iteration compute T_P and U_P:
- T_P:
  1. search in P for positive body literals, which are also in T of the current partial interpretation
  2. search in P for negative body literals, which are also in F of the current partial interpretation
  3. if a rule is satisfied add its head to T
- U_P:
  1. search in P for positive body literals, which are in F of the current partial interpretation $\Rightarrow$ rule becomes inapplicable
  2. search in P for negative body literals, which are in T of the current partial interpretation $\Rightarrow$ rule becomes inapplicable
  3. get rid of all remaining negative body literals
  4. compute the consequences $Cn(P)$ of the remaining program P
  5. the greates unfounded set U_P is the differnce of all atom $A(P)$ and $Cn(P)$
- repeat until $\Omega_P^i\langle T, F\rangle = \Omega_P^{i+1}\langle T, F\rangle$

13.3 Fill out

$$\begin{aligned}
\Omega_P^0\langle\varnothing,\varnothing\rangle &= \langle\varnothing,\varnothing\rangle\\
\Omega_P^1\langle\varnothing,\varnothing\rangle &= \langle\{a1,a2\},\{a1,a2\}\rangle\\
\Omega_P^2\langle\varnothing,\varnothing\rangle &= \langle\{a1,a2\},\{a1,a2\}\rangle\\
\Omega_P^3\langle\varnothing,\varnothing\rangle &= \langle\{a1,a2\},\{a1,a2\}\rangle\\
\Omega_P^4\langle\varnothing,\varnothing\rangle &= \langle\{a1,a2\},\{a1,a2\}\rangle\\
\bigsqcup_{i\geqslant 0}\Omega_P^i\langle\varnothing,\varnothing\rangle &= \langle\{a1,a2\},\{a1,a2\}\rangle
\end{aligned}$$

14 Nogoods

14.1 Definition

Two types of **nogoods**:

1. compuation nogoods Δ_P
   (a) atom-oriented nogoods
      i. given rules : $a \leftrightarrow v_{B_1} \vee \ldots \vee v_{B_k}$, yields:
      ii. $\Delta(a) = \{\{F_a, T_{B_1}\}, \ldots, \{F_a, T_{B_k}\}\}$
      $\rightarrow$ it cannot be the case, if a is *False*, that one of its bodies $B_1, \ldots, B_k$ is *True*.

iii. $\delta(a) = \{T_a, F_{B_1}, \ldots, F_{B_k}\}$
$\rightarrow$ it cannot be the case, if a is *True*, that all of its bodies $B_1, \ldots, B_k$ are *False*.

(b) body-oriented nogoods

i. given a body: $v_B \leftrightarrow a_1 \wedge \ldots \wedge a_m \wedge \neg a_{m+1} \wedge \ldots \wedge \neg a_n$, yields:

ii. $\Delta(B) = \{\{T_B, F_{a_1}\}, \ldots, \{T_B, F_{a_m}\}, \{T_B, T_{a_{m+1}}\}, \ldots \{T_B, T_{a_n}\}\}$
$\rightarrow$ it cannot be the case, if B is *True*, that one of its pos. body lit. $a_1, \ldots, a_m$ is *False* and one of its neg. body lit. $\neg a_{m+1}, \ldots, \neg a_n$ is *True*.

iii. $\delta(B) = \{F_B, T_{a_1}, T_{a_m}, F_{a_{m+1}}, F_{a_n}\}$
$\rightarrow$ it cannot be the case, if B is *False*, that all of its pos. body lit. $a_1, \ldots, a_m$ are *True* and all of its neg. body lit. $\neg a_{m+1}, \ldots, \neg a_n$ are *False*.

2. loop nogoods Λ_P

(a) given a loop U $\varnothing \subset U \subseteq A(P)$, the loop nogood of an atom $a \in U$ is:

(b) $\lambda(a, U) = \{T_a, F_{B_1}, \ldots, F_{B_k}\}$, where:

(c) $EB_P(U) = \{B_1, \ldots, B_K\}$ is the external support for the loop.

$\rightarrow$ it cannot be the case, if a is *True*, that all body literals of its external support are *False*.

(d) $\Lambda_P = \bigcup_{\varnothing \subset U \subseteq A(P)} \{\lambda(a, U) \mid a \in U\}$

14.2 First UIP

Derive a conflict nogood from a violated nogood, by iteratively resolving violating literals.

1. start from the violated nogood
2. look for the latest derivation of a literal at the current decision level
3. copy every literal of the nogood except the one to resolve
4. resolve the literal using the associated nogood
5. repeat until only one literal of the current decision level is part of the conflict nogood
6. undo the latest decision level and continue with the derived conflict nogood

$$
\begin{aligned}
\Delta_P = \{ \\
\{T_a, F\{\neg b, \neg d\}, F\{\neg c, e\}\}, \\
\{F_a, T\{\neg b, \neg d\}\}, \{F_a, T\{\neg c, e\}\}, \\
\{T_b, F\{\neg c, g\}\}, \\
\{F_b, T\{\neg c, g\}\}, \\
\{T_c, F\{a, \neg d\}\}, \\
\{F_c, T\{a, \neg d\}\}, \\
\{T_d, F\{\neg c\}\}, \\
\{F_d, T\{\neg c\}\}, \\
\{T_e, F\{a, \neg g\}, F\{\neg b, \neg f\}\}, \\
\{F_e, T\{a, \neg g\}\}, \{F_e, T\{\neg b, \neg f\}\}, \\
\{T_f, F\{a, d\}\}, \\
\{F_f, T\{a, d\}\}, \\
\{T_g, F\{\neg b, \neg f\}\}, \\
\{F_g, T\{\neg b, \neg f\}\}, \\
\{T\{\neg b, \neg d\}, T_b\}, \{T\{\neg b, \neg d\}, T_d\}, \\
\{F\{\neg b, \neg d\}, F_b, F_d\}, \\
\{T\{\neg c, e\}, T_c\}, \{T\{\neg c, e\}, F_e\}, \\
\{F\{\neg c, e\}, F_c, T_e\}, \\
\{T\{\neg c, g\}, T_c\}, \{T\{\neg c, g\}, F_g\}, \\
\{F\{\neg c, g\}, F_c, T_g\}, \\
\{T\{a, \neg d\}, F_a\}, \{T\{a, \neg d\}, T_d\}, \\
\{F\{a, \neg d\}, T_a, F_d\}, \\
\{T\{\neg c\}, T_c\}, \\
\{F\{\neg c\}, F_c\}, \\
\{T\{a, \neg g\}, F_a\}, \{T\{a, \neg g\}, T_g\}, \\
\{F\{a, \neg g\}, T_a, F_g\}, \\
\{T\{\neg b, \neg f\}, T_b\}, \{T\{\neg b, \neg f\}, T_f\}, \\
\{F\{\neg b, \neg f\}, F_b, F_f\}, \\
\{T\{a, d\}, F_a\}, \{T\{a, d\}, F_d\}, \\
\{F\{a, d\}, T_a, T_d\}, \\
\} \\
loop(P) = \{a, e\} \\
\Lambda_P = \{ \\
\{T_a, F\{\neg b, \neg d\}, F\{\neg b, \neg f\}\} \\
\{T_e, F\{\neg b, \neg d\}, F\{\neg b, \neg f\}\} \\
\}
\end{aligned}
$$