Technische Universität Wien
Fakultät für Informatik
Uwe Egly
Antonius Weinzierl

WS 2012
Übungsblatt Block 2
Matrikelnummer: XXXXXXXX

Exercises on Formal Methods in Computer Science

If you would like to receive feedback *in the exercise sessions*, you should submit your solutions to TUWEL no later than *November 13th 2012*. If you upload you exercises up to *November 20th 2012*, you will get feedback in electronic form.

Exercise 1 Tseitin Transformation

(a) For the formula $\psi = \big(a \to (b \to \neg a)\big)$ use Tseitin to compute a sat-equivalent CNF.

Solution:
The formula tree and the assigned labels for ψ are given in Figure 1.

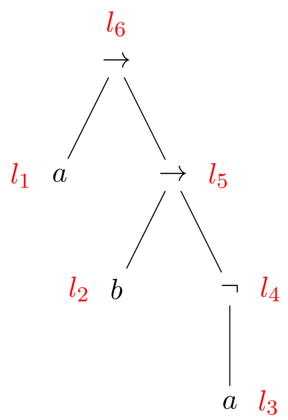


Figure 1: Formula tree for ψ and assigned labels in red.

The resulting equivalences are:

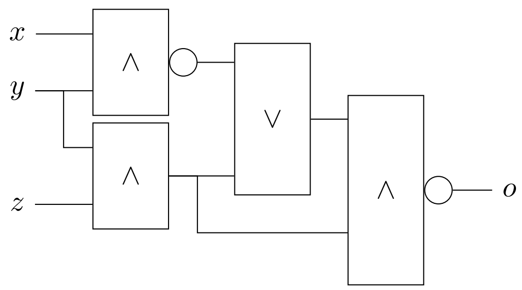
$$
\begin{aligned}
l_1 &\leftrightarrow a\\
l_2 &\leftrightarrow b\\
l_3 &\leftrightarrow a\\
l_4 &\leftrightarrow \neg l_3\\
l_5 &\leftrightarrow (l_2 \to l_4)\\
l_6 &\leftrightarrow (l_1 \to l_5)
\end{aligned}
$$

Transforming those to CNF yields:

$$
\begin{array}{lll}
\neg l_1 \vee a & l_1 \vee \neg a & \\
\neg l_2 \vee b & l_2 \vee \neg b & \\
\neg l_3 \vee a & l_3 \vee \neg a & \\
\neg l_4 \vee \neg l_3 & l_4 \vee l_3 & \\
\neg l_5 \vee \neg l_2 \vee l_4 & l_5 \vee l_2 & l_5 \vee \neg l_4 \\
\neg l_6 \vee \neg l_1 \vee l_5 & l_6 \vee l_1 & l_5 \vee \neg l_5
\end{array}
$$

If we add the single clause l_6 to the above set of clauses, then the resulting set of clauses is sat-equivalent to ψ.

(b) Given the circuit below with AND, NAND, and OR gates, use Tseitin to obtain a linear-size CNF.



Solution:

We directly label the circuit dag with labels as in Figure 2. Observe that we do not assign labels to input lines here and use NAND-gates directly (instead of decomposing them into AND followed by NOT).

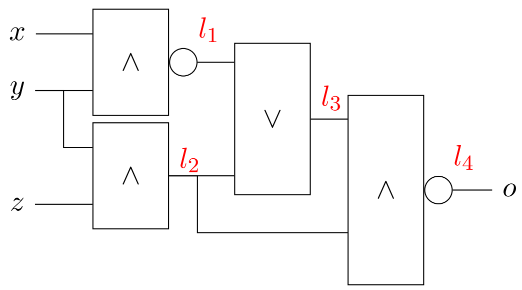


Figure 2: Labelled circuit.

So the corresponding equivalences are (where $\uparrow$ is the Sheffer stroke, a binary logical connective that is equivalent to a NAND gate, i.e., $x \uparrow y \equiv \neg(x \wedge y)$):

$$\begin{aligned}
l_1 &\leftrightarrow x \uparrow y \\
l_2 &\leftrightarrow y \wedge z \\
l_3 &\leftrightarrow l_1 \vee l_2 \\
l_4 &\leftrightarrow l_3 \uparrow l_2
\end{aligned}$$

Corresponding to those equivalences are the following clauses:

$$\begin{array}{lll}
\neg l_1 \vee \neg x \vee \neg y & l_1 \vee x & l_1 \vee y \\
\neg l_2 \vee y & \neg l_2 \vee z & l_2 \vee \neg y \vee \neg z \\
\neg l_3 \vee l_1 \vee l_2 & l_3 \vee \neg l_1 & l_3 \vee \neg l_2 \\
\neg l_4 \vee \neg l_3 \vee \neg l_2 & l_1 \vee l_3 & l_1 \vee l_2
\end{array}$$

We add the single clause l_4 to the above set and obtain a set of clauses corresponding to the above circuit, whose size is linear in the size of the circuit.

(c) Let ψ be a propositional formula and let $\hat{\delta}(\psi)$ be the set of clauses resulting from Tseitin's transformation on ψ. Prove that the following holds:

If ψ is satisfiable then $\hat{\delta}(\psi)$ is satisfiable.

You only need to prove this for the connectives $\wedge$ and $\neg$. Use the below clause schemes, which introduce a new label for every boolean variable.

$$\begin{array}{rccc}
L_a \leftrightarrow a & (\neg L_a \vee a) & (L_a \vee \neg a) & \\
L_\phi \leftrightarrow (L_1 \wedge L_2) & (\neg L_\phi \vee L_1) & (\neg L_\phi \vee L_2) & (L_\phi \vee \neg L_1 \vee \neg L_2) \\
L_\phi \leftrightarrow \neg L_1 & (\neg L_\phi \vee \neg L_1) & (L_\phi \vee L_1) &
\end{array}$$

Solution:

Let $\hat{\delta}(\psi)$ be the set of all clauses from the labelling of ψ and the additional clause (L_ψ).

We have to show: If ψ is satisfiable then $\hat{\delta}(\psi)$ is satisfiable. In other words: If there exists $I \in Mod(\psi)$ then there exists $I' \in Mod(\hat{\delta}(\psi))$, that is for every $C \in \hat{\delta}(\psi)$ holds $I'(C) = 1$.

To prove this statement, we assume that ψ is satisfiable. Then we have to show that for some interpretation I' of $\hat{\delta}(\psi)$ it holds that all clauses $C \in \hat{\delta}(\psi)$ evaluate to true, i.e., $\forall C \in \hat{\delta}(\psi) : I'(C) = 1$.

As we assumed ψ to be satisfiabel, there exists a model I of ψ. We extend I to an interpretation I' for $\hat{\delta}(\psi)$ as follows:

i) $I'(a) = I(a)$ for every propositional variable a occuring in ψ.

ii) $I'(L_\phi) = I(\phi)$ for every subformula occurrence ϕ of ψ, i.e., $\phi \in \Sigma(\psi)$, where L_ϕ is the label assigned to ϕ.

It remains to show that I' is also a model of $\hat{\delta}(\psi)$.

For the following proof we assume without further notice that ϕ is a subformula occurrence of ψ, i.e., $\phi \in \Sigma(\psi)$.

As every clause in $\hat{\delta}(\psi) \setminus \{(L_\psi)\}$ results from the translation of one subformula occurrence ϕ of ψ, we first show by structural induction on ψ that, for all $C \in \hat{\delta}(\psi) \setminus \{(L_\psi)\}$, it that holds $I'(C) = 1$. The Induction Hypothesis (IH) which we use is as follows:

IH: If ϕ' is a subformula of ϕ with $\phi' \neq \phi$ then I' satisfies all clauses in $\hat{\delta}(\psi)\setminus\{(L_\psi)\}$ that stem from the translation of ϕ'.

- Base case: $\phi = a$ where a is propositional variable. The clauses in $\hat{\delta}(\psi)$ steming from the translation of ϕ are $(\neg L_a \vee a)$ and $(L_a \vee \neg a)$. To show that they evaluate to true under I' consider all cases for $I(a)$:
  - $I(a) = 1$: then $I'(a) = 1$ by i) and $I'(L_a) = 1$ by ii), thus $I'(\neg L_a \vee a) = 1$ and $I'(L_a \vee \neg a)$.
  - $I(a) = 0$: then $I'(a) = 0$ by i) and $I'(L_a) = 0$ by ii), thus $I'(\neg L_a \vee a) = 1$ and $I'(L_a \vee \neg a)$.

  Therefore all clauses for $\phi = a$ are satisfied by I'.
- Induction step: case $\phi = \phi_1 \wedge \phi_2$. The clauses are $(\neg L_\phi \vee L_1)$, $(\neg L_\phi \vee L_2)$, $(L_\phi \vee \neg L_1 \vee \neg L_2)$ where the label for ϕ_1 is L_1, respectively for ϕ_2 is L_2.
  We consider all cases for $I(\phi)$:
  - $I(\phi) = 1$: thus $I(\phi_1) = I(\phi_2) = 1$ by the semantics of $\wedge$, so $I'(L_1) = I'(L_2) = 1$ by ii) as well as $I'(L_\phi) = 1$. Therefore $I'(\neg L_\phi \vee L_1) = I'(\neg L_\phi \vee L_2) = I'(L_\phi \vee \neg L_1 \vee \neg L_2) = 1$.
  - $I(\phi) = 0$: thus $I(\phi_1) = 0$ or $I(\phi_2) = 0$. Without loss of generality we assume $I(\phi_1) = 0$. Thus $I'(L_\phi) = I'(L_2) = 0$. Therefore $I'(\neg L_\phi \vee L_1) = I'(\neg L_\phi \vee L_2) = I'(L_\phi \vee \neg L_1 \vee \neg L_2) = 1$.

  As all clauses for ϕ_1 and ϕ_2 are satisfied by I' according IH, it follows that all clauses for $\phi = \phi_1 \wedge \phi_2$ are satisfied by I'.
- Induction step: case $\phi = \neg\phi_1$. The clauses are $(\neg L_\phi \vee \neg L_1), (L_\phi \vee L_1)$ where L_1 is the label for ϕ_1.
  We consider all cases for $I(\phi)$:
  - $I(\phi) = 1$: thus $I(\phi_1) = 0$ and by ii) is $I'(L_\phi) = 1$ and $I'(L_1) = 0$. Therefore $I'(\neg L_\phi \vee \neg L_1) = I'(L_\phi \vee L_1) = 1$.
  - $I(\phi) = 0$: thus $I(\phi_1) = 1$ and by ii) is $I'(L_\phi) = 0$ and $I'(L_1) = 1$. Therefore $I'(\neg L_\phi \vee \neg L_1) = I'(L_\phi \vee L_1) = 1$.

  As all clauses for ϕ_1 are satisfied by I' according to IH, all clauses for $\phi = \neg\phi_1$ are satisfied by I'.

The only remaining clause not covered by structural induction is (L_ψ) where L_ψ is the label assigned to ψ. As $I \in Mod(\psi)$ holds $I(\psi) = 1$ and thus by ii) holds $I'(L_\psi) = 1$.

Therefore all clauses are satisfied by I' and we have proven: if ψ is satisfiable then $\hat{\delta}(\psi)$ is satisfiable. $\square$

Shorter Alternative: One can show that the clauses for the cases $\phi = a$ and $\phi = \neg\phi_1$ evaluate to true in shorter terms. Instead of the case distinction for $I(\phi)$, directly use the relationship between ϕ and its assigned label, as shown in the following:

- Case $\phi = a$: by ii) $I'(a) = I'(L_a)$ therefore $I'(\neg L_a \vee a) = 1 - I'(L_a) + I'(L_a) = 1$ and $I'(L_a \vee \neg a) = I'(L_a) + 1 - I'(L_a) = 1$, so all clauses are satisfied.
- Case $\phi = \neg\phi_1$: by ii) and the semantics of negation it holds that $I'(L_\phi) = 1 - I'(L_1)$ therefore $I'(\neg L_\phi \vee \neg L_1) = 1 - (1 - I'(L_1)) + 1 - I'(L_1) = 1$ and $I'(L_\phi \vee L_1) = 1 - I'(L_1) + I'(L_1) = 1$. As the clauses for ϕ_1 are satisfied by IH, all clauses for ϕ are satisfied.

Notice: The rest of the proof (assumption I that is a model, induction hypothesis, etc.) remains the same.

Exercise 2 Implication Graphs

Let $\mathcal{C}$ be a clause set consisting of the following clauses:

$$\begin{aligned}
c_1: &\quad (\neg A \vee B) \\
c_2: &\quad (\neg A \vee \neg B \vee C) \\
c_3: &\quad (A \vee B) \\
c_4: &\quad (\neg F \vee \neg B \vee \neg G) \\
c_5: &\quad (G \vee \neg E) \\
c_6: &\quad (G \vee D) \\
c_7: &\quad (C \vee E \vee \neg D) \\
c_8: &\quad (\neg A \vee C)
\end{aligned}$$

(a) Draw an implication graph for $\mathcal{C}$. Use the decision $C = 0@1$, and $F = 1@2$ until you reach a conflict.

Solution:
The resulting conflict graph is givne in Figure 4.

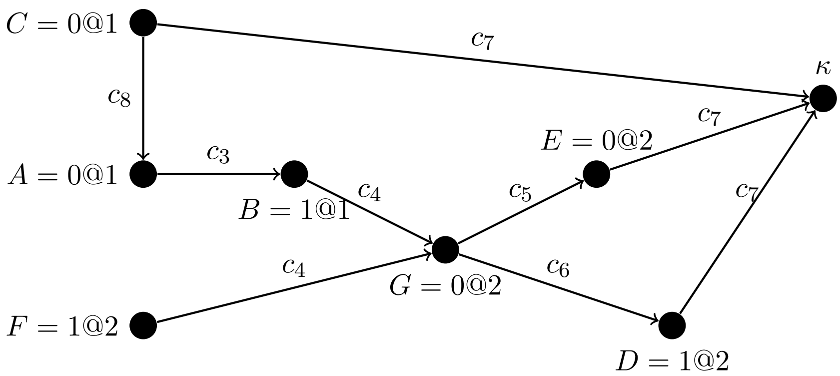


Figure 3: Implication Graph for $\mathcal{C}$ with decisions $C = 0@1$ and $F = 1@2$.

(b) Determine all UIPs in the implication graph, find the first UIP and use resolution to learn a conflict clause corresponding to the first UIP.

Solution:
UIPs (nodes through which all paths from the current decision to the conflict go through) are the nodes $F = 1@2$ and $G = 0@2$ where the latter is the first UIP (closest to the conflict).

We resolve c_7, c_5, and c_6 and obtain:

$$\begin{aligned}
r_1 := res(c_7, c_5, E) &= (C \vee G \vee \neg D) \\
r_2 := res(r_1, c_6, D) &= (C \vee G \vee G) \\
fac(r_2) &= (C \vee G)
\end{aligned}$$

So the learned clause according to the first UIP scheme is c_9: $(C \vee G)$.

(c) Add the learned clause, apply conflict-driven backtracking and draw the resulting implication graph.

Solution:
For conflict-driven backtracking, we backtrack to the second highest DL in the learned clause, i.e., we backtrack to $DL = 1$. For this kind of backtracking, we keep all decisions on $DL = 1$ but delete all others with $DL > 1$. After BCP the resulting implication graph is as in Figure 4.

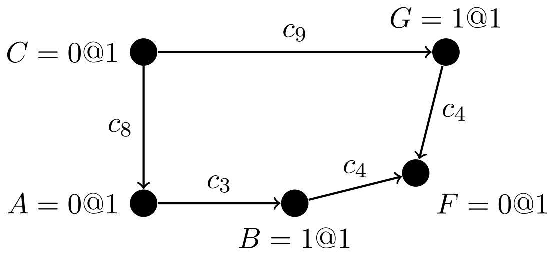


Figure 4: Implication Graph for $\mathcal{C}$ with learned clause c_9 after conflict-driven backtracking and BCP.

(d) Show that in a conflict graph the first UIP is uniquely defined, i.e., there is exactly one node in the implication graph which is a first UIP.

Solution:
Proof by contradiction: Assume there are two nodes v, v' where both v and v' are first-UIPs. Let d be the node of the last decision and k the conflict node.

A UIP is by definition a node where all paths from d to k go through. As v and v' are first-UIPs, they both are UIPs, so all paths from d to k go through v and also through v'.

Therefore there either is a path $d, \ldots, v, \ldots, v', \ldots, k$ from v to v' or there is a path from v' to v. Without loss of generality, let the path be from v to v'. As all paths from d to k go through v and v', all paths are of form $d, \ldots, v, \ldots, v', \ldots, k$, because the implication graph is acyclic.

As $v \neq v'$ the distance $d(v', k)$ between v' and k is smaller than the distance $d(v, k)$, i.e., $d(v', k) < d(v, k)$. But this contradicts the assumption that v is a first-UIP, because v' is closer to the conflict k than v.

Therefore there can be only one first UIP.

As d, the current decision node, is always a UIP, there always exists a at least one UIP, hence there also exists a UIP closest to the conflict, i.e., there exists a first UIP.

(e) Let $\mathcal{C}$ be a set of clauses and G a conflict graph with respect to $\mathcal{C}$. Prove: if a clause C_l is learned following the first-UIP scheme, then C_l is a consequence of $\mathcal{C}$.

Solution:
Consider how a new clause is learnt: Find the first-UIP u and resolve with clauses from the conflict k to u. Let $S \subseteq \mathcal{C}$ denote those clauses that occur as edge-labels in the implication graph G from the first UIP u to the conflict node k.

As C_l is learnt following the first UIP schema, there is a resolution derivation $K_1, K_2, \ldots, K_n$ of C_l from S where $K_n = C_l$ and for each K_ℓ holds: either $K_\ell \in S$ or K_ℓ is the resolvent of two K_i and K_j with $i, j < \ell$ and $1 \leq \ell \leq n$. As resolution is correct it follows that $S \models C_l$.

By monotonicity of propositional logic it then follows that $F \cup S \models C_l$ for any set of formulas F, specifically $\mathcal{C} \cup S \models C_l$. And as $S \subseteq \mathcal{C}$ it follows that $\mathcal{C} \models C_l$.

Exercise 3 Sparse Method

Apply the Sparse Method including preprocessing on the formula φ below to obtain a propositional formula. Note that φ is not yet in NNF (Negation Normal Form).

$$(x_1 = x_2 \rightarrow x_2 = x_3) \land \big[\neg(x_2 = x_4 \lor x_3 \neq x_4 \lor x_4 \neq x_5) \lor (x_6 \neq x_5 \land x_6 = x_7 \land x_7 = x_3)\big]$$

Solution:
In the first step, we transform φ into NNF. We substitute $\rightarrow$ and apply DeMorgan to obtain φ^E, which now is in NNF:

$$(x_1 \neq x_2 \lor x_2 = x_3) \land \big[(x_2 \neq x_4 \land x_3 = x_4 \land x_4 = x_5) \lor (x_6 \neq x_5 \land x_6 = x_7 \land x_7 = x_3)\big]$$

Then, we draw the equality graph $G^E(\varphi^E)$ of φ^E, given in Figure 5. Dashed lines represent equality edges while solid lines represent disequality edges.

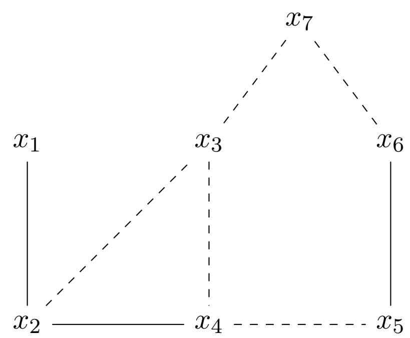


Figure 5: Equality graph $G^E(\varphi^E)$, dashed lines represent equality, solid lines disequality.

The edge (x_1, x_2) is not part of a simple contradictory cycle, therefore we set it to true and obtain φ_2^E:

$$(true \lor x_2 = x_3) \land \big[(x_2 \neq x_4 \land x_3 = x_4 \land x_4 = x_5) \lor (x_6 \neq x_5 \land x_6 = x_7 \land x_7 = x_3)\big]$$

Propositional simplification yields φ_3^E:

$$\big[(x_2 \neq x_4 \land x_3 = x_4 \land x_4 = x_5) \lor (x_6 \neq x_5 \land x_6 = x_7 \land x_7 = x_3)\big]$$

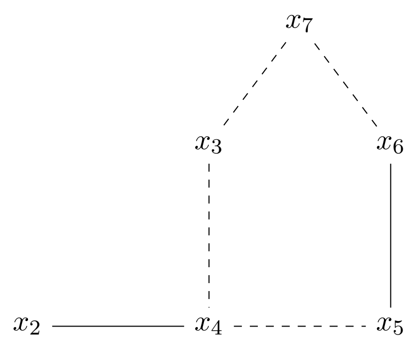


Figure 6: Equality graph $G^E(\varphi_3^E)$, dashed lines represent equality, solid lines disequality.

The equality graph $G^E(\varphi_3^E)$ then is as shown in Figure 6.
Edge (x_2, x_4) now is not in a simple contradictory cycle, therefore we set it to true and apply propositional simplification to obtain φ_4^E:

$$\big[(x_3 = x_4 \wedge x_4 = x_5) \vee (x_6 \neq x_5 \wedge x_6 = x_7 \wedge x_7 = x_3)\big]$$

The equality graph $G^E(\varphi_4^E)$ is given in Figure 7.

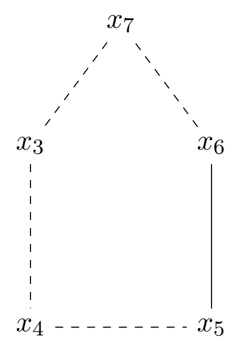


Figure 7: Equality graph $G^E(\varphi_4^E)$, dashed lines represent equality, solid lines disequality.

All edges of $G^E(\varphi_4^E)$ are part of a simple contradictory cycle, so we stop with preprocessing and build the propositional skeleton $e(\varphi_4^E)$:

$$(e_{3,4} \wedge e_{4,5}) \vee (\neg e_{5,6} \wedge e_{6,7} \wedge e_{3,7})$$

For transitivity contraints B_t we make the nonpolar equality graph $G_{NP}^E(\varphi_4^E)$ chordal as shown in Figure 8. Observe that edges (x_4, x_7) and (x_5, x_7) are introduced.
The according transitivity constraints B_t are then:

$$\begin{aligned}
&(e_{3,4} \wedge e_{4,7} \rightarrow e_{3,7}) \wedge (e_{4,7} \wedge e_{3,7} \rightarrow e_{3,4}) \wedge (e_{3,7} \wedge e_{3,4} \rightarrow e_{4,7}) \wedge \\
&(e_{4,5} \wedge e_{5,7} \rightarrow e_{4,7}) \wedge (e_{5,7} \wedge e_{4,7} \rightarrow e_{4,5}) \wedge (e_{4,7} \wedge e_{4,5} \rightarrow e_{5,7}) \wedge \\
&(e_{5,6} \wedge e_{6,7} \rightarrow e_{5,7}) \wedge (e_{6,7} \wedge e_{5,7} \rightarrow e_{5,6}) \wedge (e_{5,7} \wedge e_{5,6} \rightarrow e_{6,7})
\end{aligned}$$

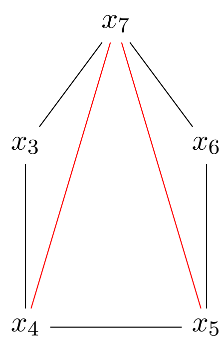


Figure 8: Nonpolar equality graph $G^E_{NP}(\varphi^E_4)$, made chordal by additional edges (in red).

The resulting formula in propositional logic then is $e(\varphi^E_4) \wedge B_t$.

<u>Exercise 4</u> Ackermann's Reduction

Apply Ackermann's reduction on the following EUF-formula φ to obtain an E-formula:

$$F(F(x_1)) \neq F(x_1) \wedge G(x_1, x_2) = F(x_2) \wedge F(G(x_2, F(x_2))) \neq F(F(x_1))$$

Solution:
We first number the instances of the UFs inwards-to-outwards, left-to-right:

$$F_2(F_1(x_1)) \neq F_1(x_1) \wedge G_1(x_1, x_2) = F_3(x_2) \wedge F_4(G_2(x_2, F_3(x_2))) \neq F_2(F_1(x_1))$$

This already gives $\mathcal{T}$ for the numbered instances. For example:

$$\begin{aligned}
\mathcal{T}(F_1(x_1)) &= f_1 \\
\mathcal{T}(F_2(F_1(x_1))) &= f_2 \\
\mathcal{T}(F_3(x_2)) &= f_3 \\
\mathcal{T}(F_4(G_2(x_2, F_3(x_2)))) &= f_4 \\
\mathcal{T}(G_1(x_1, x_2)) &= g_1 \\
\mathcal{T}(G_2(x_2, F_3(x_2))) &= g_2
\end{aligned}$$

So $flat^E := f_2 \neq f_1 \wedge g_1 = f_3 \wedge f_4 \neq f_2$.
Based on $\mathcal{T}$ we construct $FC^E :=$

$$\begin{aligned}
(x_1 = f_1 &\rightarrow f_1 = f_2) \wedge \\
(x_1 = x_2 &\rightarrow f_1 = f_3) \wedge \\
(x_1 = g_2 &\rightarrow f_1 = f_4) \wedge \\
(f_1 = x_2 &\rightarrow f_2 = f_3) \wedge \\
(f_1 = g_2 &\rightarrow f_2 = f_4) \wedge \\
(x_2 = g_2 &\rightarrow f_3 = f_4) \wedge \\
((x_1 = x_2 \wedge x_2 = f_3) &\rightarrow g_1 = g_2)
\end{aligned}$$

Finally $\varphi^E := FC^E \rightarrow flat^E$.