Assessed Coursework: Systems Verification

Ioannis Kassinopoulos

February 23, 2013

Question 1

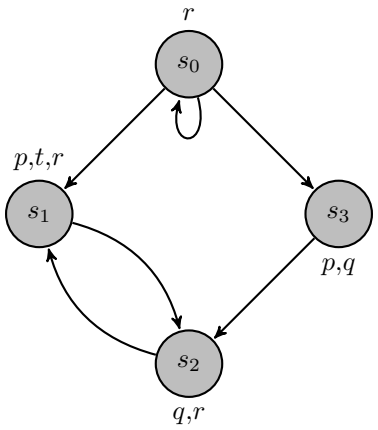

Figure 1: The transition system $\mathcal{M}_1$.

Algebraic Form

A transition system $\mathcal{M} = (S, \rightarrow, \pi)$ is a set of states S endowed with a transition relation $\rightarrow$ (a binary relation on S), such that every $s \in S$ has some $s' \in S$ with $s \rightarrow s'$, and an inverse labeling function $\pi : \mathcal{P} \rightarrow S$.

Our system $\mathcal{M}_1$ (figure: 1) can be described as following:
$\mathcal{P} = \{p, q, r, t\}$
$\mathcal{M}_1 = \{\{s_0, s_1, s_2, s_3\}, \{(s_0, s_0), (s_0, s_1), (s_0, s_3), (s_1, s_2), (s_2, s_1), (s_3, s_1)\}, \pi\}$
$\pi(p) = \{s_1, s_3\}$
$\pi(q) = \{s_2, s_3\}$
$\pi(r) = \{s_0, s_1, s_2\}$
$\pi(t) = \{s_1\}$

Infinite Tree

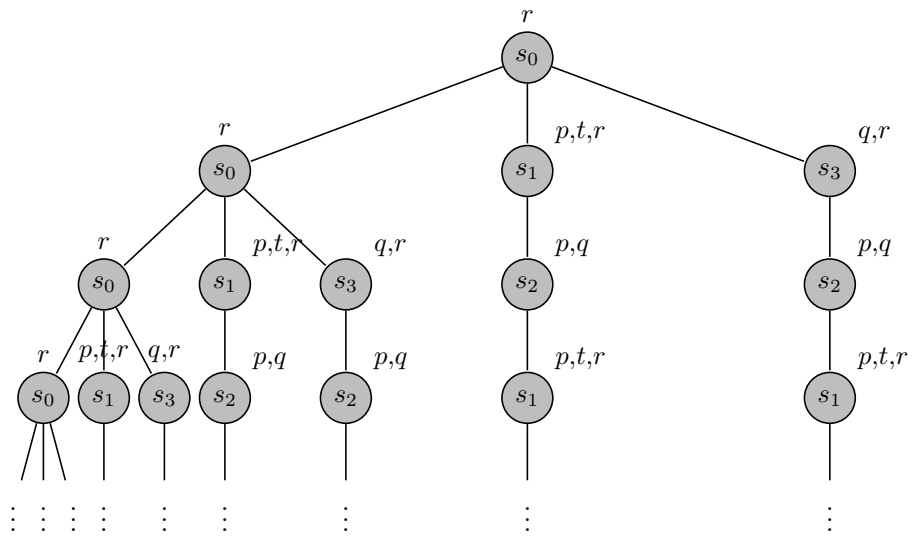

Figure 2: Unwinding the system described by $\mathcal{M}_1$ as an infinite tree of all computation paths beginning in s_0 (first layer).

Satisfiability

$\phi = Ft$

$(\mathcal{M}_1, s_0) \not\models Ft$

since for path $\rho = x_0, x_1, ... = s_0^+$ no state $x_i \notin \pi(t)$ since $s_0 \notin \pi(t)$

$(\mathcal{M}_1, s_2) \models Ft$

since the only path that exist is $\rho = x_0, x_1, ... = (s_2, s_1)^+$ and for $x_i = s_1 \Rightarrow s_1 \in \pi(t)$

$\phi = \neg EGr$

$EGr = \neg AF\neg r \Rightarrow \neg EGr = \neg\neg AF\neg r = AF\neg r$

$(\mathcal{M}_1, s_0) \not\models AFr$ since for the path $\rho = x_0, x_1, ... = s_0^+$ every state $x_i = s_0 \in \pi(r)$

$\Rightarrow (\mathcal{M}_1, s_0) \not\models \neg EGr$

$(\mathcal{M}_1, s_2) \not\models AFr$ since for the only path $\rho = x_0, x_1, ... = (s_2, s_1)^+$ every state $x_i = s_1 \in \pi(r)$ or $x_i = s_2 \in \pi(r)$

$\Rightarrow (\mathcal{M}_1, s_2) \not\models \neg EGr$

$\phi = E(tUq)$

$(\mathcal{M}_1, s_0) \not\models E(tUq)$

since for every (and therefore for at least one) path $\rho = x_0, x_1, ... = s_0, ...$ at s_0, $s_0 \notin \pi(t) \cup \pi(q)$.

therefore, since at the initial state we have neither t nor q we cannot say that a path exists starting from s_0 such that t until q holds.

$(\mathcal{M}_1, s_2) \models E(tUq)$

since for the only path $\rho = x_0, x_1, ... = (s_2, s_1)^+$ we have at $x_0 = s_2$, and $(\mathcal{M}_1, s_2) \models tUq$ since $s_2 \in \pi(q)$. This means that t is always true up to the point that q gets true.

$\phi = E(FGp)$

For this formula we first need to consider all the possible tuples of transition relation given by the $\rightarrow$ set:

$\{(s_0, s_0), (s_0, s_1), (s_0, s_3), (s_1, s_2), (s_2, s_1), (s_3, s_1)\}$

As we can see from the $\pi(p)$ set, no two concequtive states $x_i, x_{i+1} \in \pi(p)$ which means that for every state s_i: $(\mathcal{M}_1, s_i) \not\models Gp$, $(\mathcal{M}_1, s_i) \not\models FGp$ and $(\mathcal{M}_1, s_i) \not\models E(FGp)$

Therefore we can say that:

$(\mathcal{M}_1, s_0) \not\models E(FGp)$

$(\mathcal{M}_1, s_2) \not\models E(FGp)$

since no path exists which eventually gets forever p starting from either s_0 or s_2

$\phi = EGr$

$EGr = \neg\neg EGr$ therefore from the $\phi = \neg EGr$ solution we can deduce that:

$(\mathcal{M}_1, s_0) \models EGr$ since $(\mathcal{M}_1, s_0) \not\models \neg EGr$ since $true = \neg false$

and

$(\mathcal{M}_1, s_2) \models EGr$ since $(\mathcal{M}_1, s_2) \not\models \neg EGr$ since $true = \neg false$

$\phi = G(r \vee q)$

From the model we can see that the following are true:

$s_0 \in \pi(r) \cup \pi(q)$ since $s_0 \in \pi(r)$ so $(\mathcal{M}_1, s_0) \models r \vee q$

$s_1 \in \pi(r) \cup \pi(q)$ since $s_1 \in \pi(r)$ so $(\mathcal{M}_1, s_1) \models r \vee q$

$s_2 \in \pi(r) \cup \pi(q)$ since $s_2 \in \pi(q)$ so $(\mathcal{M}_1, s_2) \models r \vee q$

$s_3 \in \pi(r) \cup \pi(q)$ since $s_3 \in \pi(q)$ so $(\mathcal{M}_1, s_3) \models r \vee q$

We can therefore say that for every state s_i, $(\mathcal{M}_1, s_i) \models G(r \vee q)$ since every path, from every state, will satisfy $r \vee q$ forever (globally).

$(\mathcal{M}_1, s_0) \models G(r \vee q)$

$(\mathcal{M}_1, s_2) \models G(r \vee q)$

$\phi = falseUp$

$(\mathcal{M}_1, s_0) \not\models falseUp$ since $s_0 \notin \{\} \cup \pi(p)$ therefore since the first state of the path does not fulfill p

$\Rightarrow$ false until p cannot be true.

$(\mathcal{M}_1, s_2) \not\models falseUp$ since $s_2 \notin \{\} \cup \pi(p)$ therefore since the first state of the path does not fulfill p

$\Rightarrow$ false until p cannot be true.

$\phi = A(pU(EFq))$

First, let's find which states satisfy EFq

$\{s_0, s_1, s_2, s_3\}$ satisfy EFq since all of them are either labelled q

or have one of their next states satisfying q.

This intuitively says for all states s_i, $(\mathcal{M}_1, s_i) \models EFq$.

This means that for all paths starting from every state of our model $pUEFq$ is true since we don't care about p being true since EFq is always true at the first (initial) state.

We can therefore deduce from the above that for all states s_i, $(\mathcal{M}_1, s_i) \models A(pUEFq)$ and:

$(\mathcal{M}_1, s_0) \models A(pUEFq)$

$(\mathcal{M}_1, s_2) \models A(pUEFq)$

Question 2

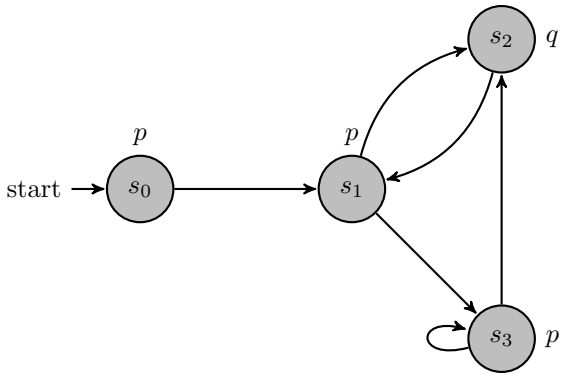

Figure 3: The transition system $\mathcal{M}_2$.

Calculating

$\phi = p$

$SAT(p) = \{s \in S \mid p \in L(s)\} = \{s_0, s_1, s_2\}$

$\Rightarrow [\![\phi]\!] = \{s_0, s_1, s_2\}$

$\phi = AGEFp$

$SAT(EFp) = SAT(E[trueUp]) = SAT_{eu}(true, p)$

$SAT_{eu}(true, p)$

$W = \{s_0, s_1, s_2, s_3\}$

$Y_0 = \{s_0, s_1, s_3\}$

$Y_1 = \{s_0, s_1, s_2, s_3\}$

$Y_2 = Y_1$

$SAT(AGEFp) = SAT(\neg EF \neg EFp)$

$SAT(EF \neg EFp) = SAT(E[trueU \neg EFp]) = SAT_{eu}(true, \neg EFp)$

$SAT_{eu}(true, \neg EFp)$

$W = \{s_0, s_1, s_2, s_3\}$

$Y_0 = S \backslash \{s_0, s_1, s_2, s_3\} = \{\}$

$Y_1 = Y_0$

$SAT(AGEFp) = SAT(\neg EF \neg EFp) = S \backslash SAT(EF \neg EFp) = \{s_0, s_1, s_2, s_3\}$

$\Rightarrow [\![\phi]\!] = \{s_0, s_1, s_2, s_3\}$

$\phi = AFq$

$SAT(AFq) = SAT_{af}(q)$

$SAT_{af}(q)$

$Y_0 = \{s_2\}$

$Y_1 = Y_0$

$\Rightarrow [\![\phi]\!] = \{s_2\}$

$\phi = AGp \vee Afq$

$SAT(AGp) = SAT(\neg EF \neg p)$

$SAT(EF \neg p) = SAT(E[trueU \neg p]) = SAT_{eu}(true, \neg p)$

$SAT_{eu}(true, \neg p)$

$W = \{s_0, s_1, s_2, s_3\}$

$Y_0 = S \backslash SAT(p) = \{s_2\}$

$Y_1 = \{s_1, s_2, s_3\}$

$Y_2 = \{s_0, s_1, s_2, s_3\}$

$Y_3 = Y_2$

$SAT(\neg EF \neg p) = S \backslash SAT(EF \neg p) = \{\}$

$SAT(AGp \vee Afq) = SAT(AGp) \cup SAT(AFq) = \{\} \cup \{s_2\} = \{s_2\}$

$\Rightarrow [\![\phi]\!] = \{s_2\}$

$\phi = E(pU(AFq))$

$SAT(E[pU(AFq)]) = SAT_{eu}(p, AFq)$

$SAT_{eu}(p, AFq)$

$W = SAT(p) = \{s_0, s_1, s_3\}$

$Y_0 = SAT(AFq) = \{s_2\}$

$Y_1 = \{s_1, s_2, s_3\}$

$Y_2 = \{s_0, s_1, s_2, s_3\}$

$Y_3 = Y_2$

$\Rightarrow [\![\phi]\!] = \{s_0, s_1, s_2, s_3\}$

Question 3

Question 4

Let $\phi = (x_1 \wedge x_2) \vee (y_1 \wedge y_2)$, the following truth table is derived to help us with our calculations

x_1	x_2	y_1	y_2	$x_1 \wedge x_2$	$y_1 \wedge y_2$	$(x_1 \wedge x_2) \vee (y_1 \wedge y_2)$
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	1	1
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	1	1	1

Binary Decision Tree

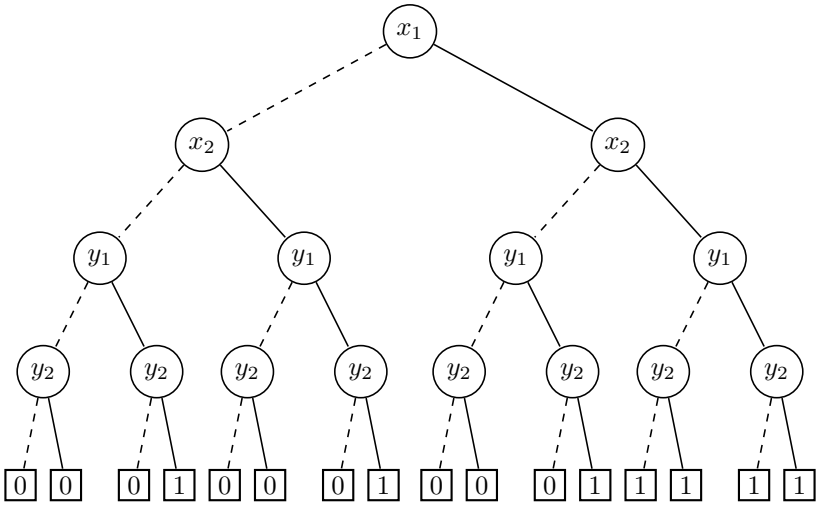

Figure 4: A BDT is easily derived from the truth table. Every non-terminal node is labelled with a variable and every terminal node is labelled with either 0 or 1.

Reduced Ordered Binary Decision Diagrams

In order to reduce the size of the BDT we can produce a Binary Decision Diagram which is a reduced form of the BDT. Making this diagram ordered over a list of variables, results in getting an Ordered Binary Decision Diagram (OBDD) which is then unique when it is reduced until no more reduction can occur. This reduced form is called canonical form and it can be used to extract equivalences since two different but equivalent Boolean functions always have identically structured Reduced Ordered Binary Decision Diagrams if they have compatible variable orderings.

Reduction Algorithm

In order to reduce BDTs we use iteratively the rules C1-C3 until no more reductions can occur.

- **C1:** Removal of duplicate terminals.

- **C2:** Removal of redundant tests.
- **C3:** Removal of duplicate non-terminals.

ROBDD under the $[x_1, x_2, y_1, y_2]$ ordering.

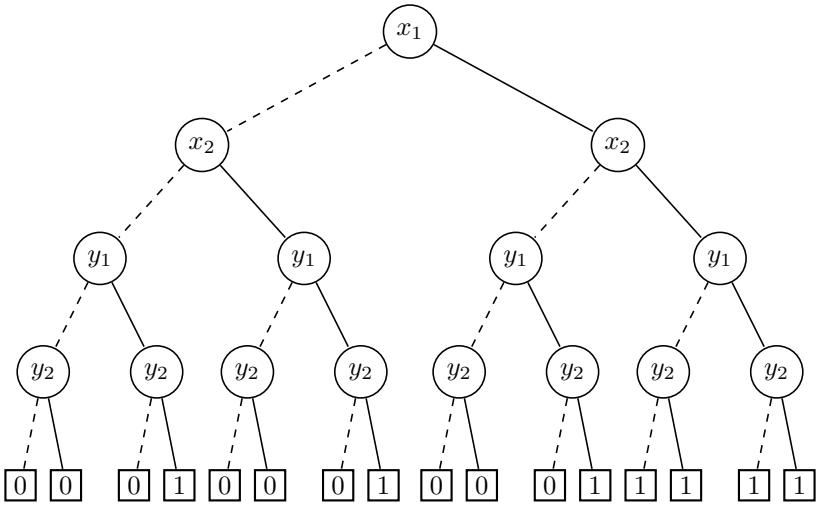

Figure 5: We start with the BDT over our ordering.

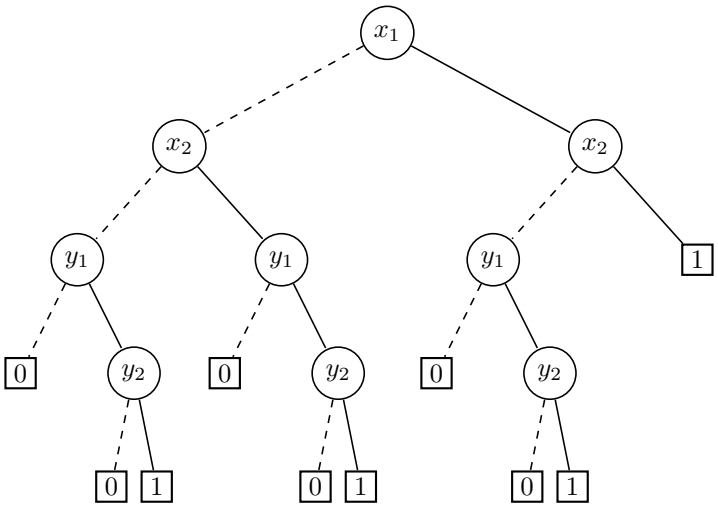

Figure 6: Using C2 we remove the redundant tests and eliminate the nodes leading to them. We derive the above reduced diagram.

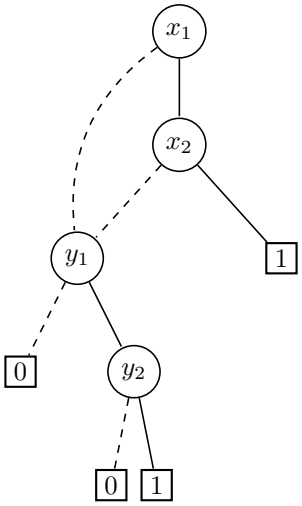

Figure 7: Using C3 we remove the duplicate non-terminals, redirect the incoming edges and derive the above reduced diagram.

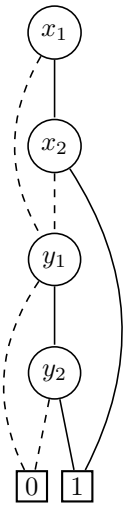

Figure 8: Using C1 we remove all the duplicate terminals. This OBDD cannot be reduced any further so we can now call it the canonical of the previous diagrams.

ROBDD under the $[x_1, y_1, y_2, x_2]$ ordering.

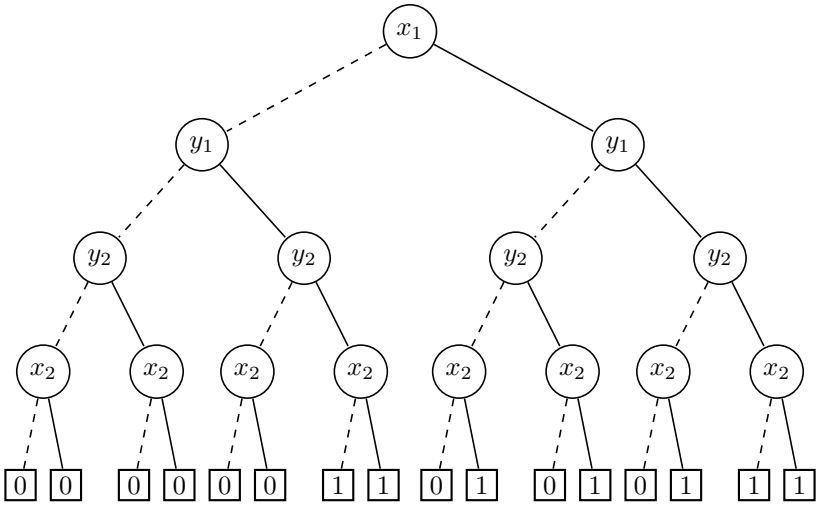

Figure 9: We start with the BDT over our ordering.

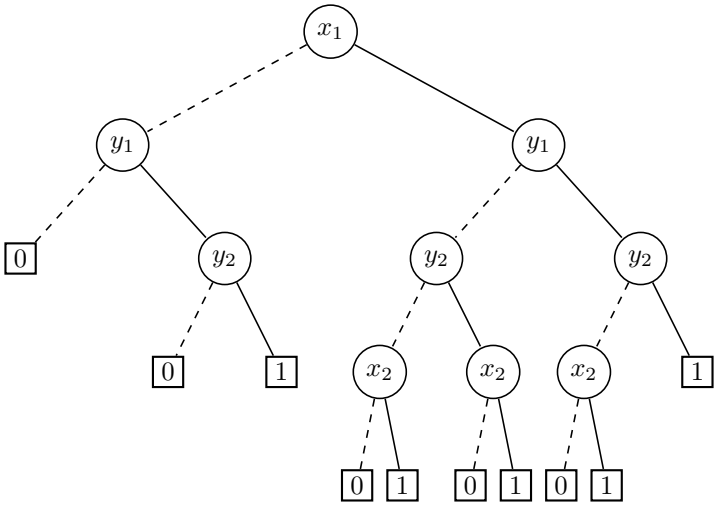

Figure 10: Using C2 we remove the redundant tests and eliminate the nodes leading to them. We derive the above reduced diagram.

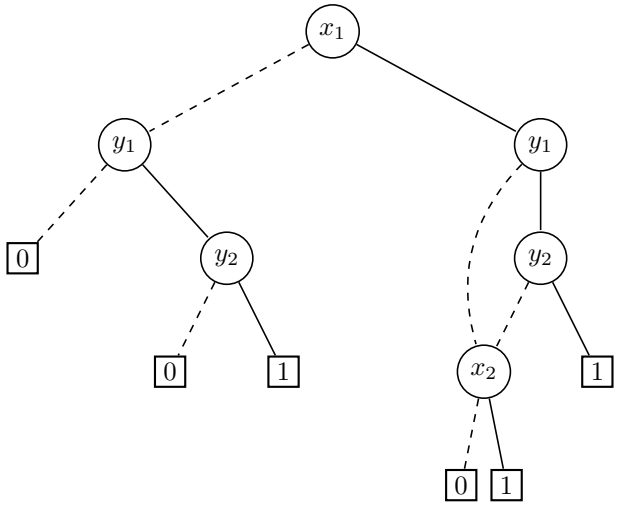

Figure 11: Using C3 we remove the duplicate non-terminals, redirect the incoming edges and derive the above reduced diagram.

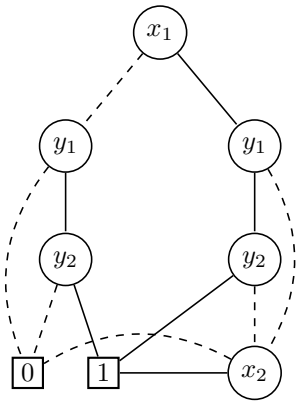

Figure 12: Using C1 we remove all the duplicate terminals. This OBDD cannot be reduced any further so we can now call it the canonical of the previous diagrams.

Discuss how ordering impacts ROBDD

Suggest an algorithm for choosing ordering