CS226 Quiz 3 (Spring 2016)

April 11, 2016 **Time: 90 mins**

- *Be brief, complete and stick to what has been asked.*
- *Unless asked for explicitly, you may cite results/proofs covered in class without reproducing them.*
- *If you need to make any assumptions, state them clearly.*
- ***Please start writing your answer to each sub-question on a fresh page. DO NOT write answers to multiple sub-questions on the same page.***
- ***The use of internet enabled devices is strictly prohibited. You will be debarred from taking the examination if you are found accessing the internet during the examination.***
- ***Please do not engage in unfair or dishonest practices during the examination. Anybody found indulging in such practices will be referred to the D-ADAC.***

1. Consider the structurally hashed AIG shown in Fig. 1. This AIG represents two functions F and G, with shared AIG nodes.

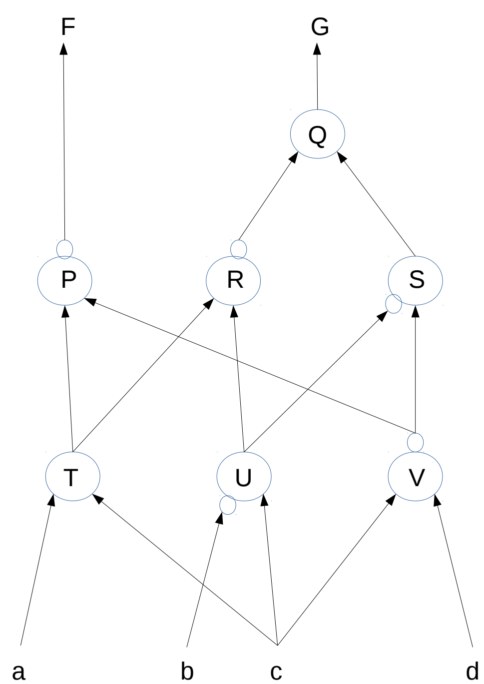

Figure 1: And-inverter-graph

We wish to compute the AIGs corresponding to the cofactors $F|_{a=1}$ and $G|_{a=1}$, i.e. the cofactors of F and G with respect to $a = 1$. In order to do this, we will use the following three (recursive) rules, where φ, φ_1 and φ_2 represent arbitrary Boolean formulas:

R0: $a|_{a=1} \; = \; 1.$

R1: $(\mathsf{NOT}\ \varphi)|_{a=1} \; = \; \mathsf{NOT}\ (\varphi|_{a=1})$

R2: $(\varphi_1\ \mathsf{AND}\ \varphi_2)|_{a=1} \; = \; (\varphi_1|_{a=1})\ \mathsf{AND}\ (\varphi_2|_{a=1})$

You are required to recursively apply these rules, and indicate the steps as shown in the table below. To get you started, the first few rows of the table are filled out, showing how the recursive rules are applied when trying to find the AIG for $F|_{a=1}$ and for $G|_{a=1}$.

The result of this recursive computation will give rise to an AIG. You are required to show the resulting AIG, which must have the AIGs for F and G (as shown in Fig. 1) embedded in it. Thus, the resulting AIG should have four roots, one each for F, G, $F|_{a=1}$ and $G|_{a=1}$.

IMPORTANT: The resulting AIG you create MUST be structurally hashed, and MUST NOT be simplified using any rewrite rules. Any simplification using rewrite rules will fetch 0 marks. Please be warned that this means you cannot simply compute $F|_{a=1}$ and $G|_{a=1}$ independently, and draw their AIGs separately.

Suggested table for showing your computation:

Cofactor	=	Recursive Cofactors	Label of node/root representing result (after structural hashing)	Rule Applied
$F\|_{a=1}$	=	$\mathsf{NOT}\ (P\|_{a=1})$	New root	R1
$P\|_{a=1}$	=	$(T\|_{a=1})\ \mathsf{AND}\ ((\mathsf{NOT}\ V)\|_{a=1})$	Node ...	R2
$((\mathsf{NOT}\ V)\|_{a=1})$	=	$\mathsf{NOT}\ (V\|_{a=1})$	Node ...	R1
$\vdots$	=	$\vdots$	$\vdots$	$\vdots$
$G\|_{a=1}$	=	$((\mathsf{NOT}\ R)\|_{a=1})\ \mathsf{AND}\ (S\|_{a=1})$	New root	R2
$\vdots$	=	$\vdots$	$\vdots$	$\vdots$

Please label new nodes in your AIG using upper-case letters, if needed, when filling up the above table.

2. Consider the circuit obtained by replacing each AIG node in Fig. 1 with a two-input AND gate, and each inverting bubble with a NOT gate. Suppose further that the delay of each NOT gate is 1 time unit, and the delay of each two-input AND gate is 2 time units.

   Topologically, the minimum delay from any input to any output of this circuit is 5 time units ($a \to T \to P \to F$), while the maximum delay from any input to any output is 8 time units ($b \to U \to R \to Q \to G$) or ($b \to U \to S \to Q \to G$).

   For purposes of this question, we will hold c and d at constant values and change both a and b simultaneously at time $t = 0$. Under these circumstance, we want to ask if it is possible to have F transition **only once** at a time **later than** $t = 5$, and also have G transition **only once** at a time **earlier than** $t = 8$.

   If so, indicate what constant values should be held at c and d, and what transitions (0-to-1 or 1-to-0) should be applied to a and b at $t = 0$ for this to happen. You must also indicate the paths along which the transitions propagate from the inputs to the outputs for the outputs to transition as required, and give the waveforms at outputs of gates along these paths.

   If the above is not possible, indicate clearly why this is so.

   Answers without justification will fetch 0 marks.