**LSV PA2 Report** R09943100 黃友杉

Implementation:

首先以Abc\_NtkForEachPo遍歷每個PO，對每個PO以Abc\_NtkCreateCone得到對應的TFI cone。因為使用TFI cone，為了最後正確輸出資訊，要先記錄cone的PI如何對應到原Ntk的PI，此作業中我利用node的name來取得及記錄cone的PI編號如何對應到Ntk PI編號。

得到TFI cone後以Abc\_NtkToDar將它轉為aig並以Cnf\_Derive將之轉為一份後續將作為pos part的cnf，接著以Cnf\_DataDup得到另一份後續將作為neg part的cnf，其中pos part指的是輸入將設為1的部分，neg part指的是輸入將設為0的部分。對於neg part，使用Cnf\_DataLift使兩者的variable不重複。至此，兩份aig電路的cnf已經準備好。

接著，以sat\_solver\_new初始化sat solver，並將兩部分電路的cnf以sat\_solver\_addclause加到solver的clauses裡。因為要用incremental sat solving來依序判斷每個PI的unate資訊，所以對每個PI要在cnf中設一個變數用來控制相關的clauses要不要enable。相關的clauses指的是對第i個PI, PIi，建立αi以及加入以下clauses: 。當時，必須整個cnf才有可能sat，因此，當我們想判斷第i個PI時，將設為0，其它設為1，這樣可以促使除了關心的那個PI外，其它PI在兩份電路的輸入必須相同。加入αi的方法為呼叫sat\_solver\_addvar。至此，cnf需要的clauses已被加入完畢。

接著需要設定sat solving的assumption。當我們想判斷該PI是否為positive unate時，須將pos part的輸出設為0，neg part輸出設為1，因為若這樣unsat，代表f(…,0,…)→f(…,1,…)，即為postive unate，反之若pos part的輸出設為0，neg part輸出設為1時unsat，則為negative unate。對於設定這些assumption的部分，αi可由加入它們到solver時sat\_solver\_addvar的回傳值知道相對應的variable編號，利用toLitCond可設定alpha的phase。PO, PI的variable編號可用pVarNums[Aig\_ObjId(node)]取得。

分別以兩種assumptions solve後，可以知道該PI為何種unate。對於不是該PO的support的PI，由定義知道它們同時是positive unate及negative unate。最後，利用前面記錄的cone的PI與原Ntk的PI的對應關係，輸出所需資訊。