

(1)

BuildBdd: 利用已寫好的函數來取得每個 AIG_Node 的兩個 input，確認他們有無 complement，再把他們 AND 起來。

BuildNtkBdd: 由於 PI 和 FF 在 SetOrder 都已經處理完，所以就把所有 AIG_Node 的 Bdd 都建好即可。

BuildPInitialState(): $\text{init} = a'b'c'd'e'd'f' \dots\dots\dots$
 $\rightarrow \sim\text{init} = a+b+c+d+e+f+\dots\dots\dots$
 $_InitialState = \sim(\sim\text{init})$
 利用這樣就能算出 $_InitialState$ 的 Bdd。

BuildPTransRelation(): $\text{tri}(Y,X,I) = (y1=F1(X,I)(y2=F2(X,I)(y3=F3(X,I)(y4=F4(X,I) \dots\dots\dots$
 $\rightarrow \sim\text{tri}(Y,X,I) = (y1 \wedge F1(X,I) + (y2 \wedge F2(X,I) + (y3 \wedge F3(X,I) + (y4 \wedge F4(X,I) \dots\dots\dots$
 $_Tri(Y,X,I) = \sim(\sim\text{tri}(Y,X,I))$
 $_Tr(Y,X)$ 就利用 $_Tri(Y,X,I)$ 和 exist 函數即可取得。

BuildPImage(): 主要利用 getPReachState() 函數來實做，
 $\text{state} = _tr \ \& \ \text{getPReachState}()$
 如果過程中發現 $\text{state} = \text{getPReachState}()$ ，就表示 Fixed point 到了。
 $_reachStates$ 是個 vector，所以用 push_back 來加入 state。

RunPCheckProperty: 主要看 $\text{monitor} \& \text{getPReachState}()$ 是否等於 constant zero，
 是的話就代表 monitor 是安全的。
 不是的話就代表有 counter example，
 代入所有 input 可能以找出所有 counter example。

(2)

```
assign x= initialized && (serviceTypeOut == `SERVICE_ON) &&
      (itemTypeOut != `ITEM_NONE);
      //On state 時 itemtype 一定會是 ITEM_NONE
assign y= initialized && (serviceTypeOut == `SERVICE_BUSY) && (inputValue>2'd3);
      //inputValue 不行超過 3
assign z= initialized && (serviceTypeOut == `SERVICE_OFF) &&
      (serviceTypeOut == `SERVICE_ON);
      //不可能同時處在 On state 與 Off state
```

(3)

驗證結果 monitor x,y,z 皆是安全的。

(4)

兩者在 ptr 指令的運行時間都超久，整體感覺 ref 比我的快一點。

(5)

abstraction 就是把 vending 的 item_type, coin_type, item_cost 種類減少。