第五章 使用 ArCats 檢驗系統性質實驗結果分析

本章我們將展示我們利用ArCats和一般局部性分析（CRA）工具在檢驗系統性質上的差異，並確信利用CCS-based模型來檢驗系統性質時，能達到如同Shi-Chi Cheung以CSP-based模型來檢驗系統性質時相同的檢驗優勢。5.1中利用加油站系統來比較在檢驗安全性質上的差異。5.2則利用RMTP系統去比較活化性質上的差異。

5.1 ArCats 和 CRA 在檢驗安全性質的結果分析

如之前所提到的，局部性分析的成功關鍵在於有良好的階層式架構和盡量將子系統中的內部行為 τ 透過最小化的方式來減緩組態爆炸。在第一個關鍵中我們可以利用 ArCats 中的 Model architecture Refactoring 功能，將系統中原本不良的階層架構轉換成適合局部性分析的階層架構;而在第二個關鍵中我們利用 Branch Bisimulation 來達到最小化。事實上在檢測安全性質時，我們必須在適當的階層中使用 Branch Bisimulation，才能確保安全性質不會因為最小化的結果而消失掉。

所謂適當的階層是指所有的欲檢驗安全性質都已經完成檢驗。簡略的說，當 export set 中不存在任何將檢驗的安全性質 action 時，我們可以執行 Branch Bisimulation。利用局部性分析檢驗安全性質並同時進行最小化的功能其步驟如下：

（1） 首先在適當的階層加入欲檢驗的安全性質，在此階層的子系統中將展開可能的死結狀態，而子系統是以 buttom-up 的方式由下往上結

合成全域系統（global system）。

（2） 當子系統在做同步的溝通中，如果進入死結狀態，我們將其轉移行動標記為τ_s(Stau)，τ_s是一種特別的內部行為，不會在最小化的過程中消失。

（3） 當合成的子系統中的 export set 中不存在任何安全性質的行動時，都可以執行 Branch Bisimulation 來達到最小化的結果。

在傳統的檢驗安全性質時，將保留部分行動到全域系統中。這樣做的目的是為了避免經過最小化的動作後，使欲檢驗的系統性質被破壞掉，不過保留全域行動將使子系統中的狀態個數相對的增加許多。在我們的系統中利用步驟(2)的方法，將不需保留額外的全域行動在系統中。事實上，局部性分析在平行合成子系統時，如果沒有發現死結時，將可以透過最小化來減緩組態爆炸的發生；如果展開後的狀態確實擁有死結狀態時，由於在每個子系統中只保留最少的全域行動τ_s，這樣也會減緩組態爆炸的發生。

表格 5-1 比較 CRA 和 ArCats_CRA 在檢驗安全性質 Right_change 時的差異

使用方法 / 子系統	**CRA**_RIGHT_CHANGE		**ARCATS_CRA**_RIGHT_CHANGE			
	Time(sec)	**State#**	**Time(sec)**	**State#**	**Time(sec)**	**Min. State#**
Counter	0.046	25	0.046	25	0.046	25
Station	0.03	27	0.03	27	0.03	27
Client	0.015	16	0.015	16	0.015	16
Gas-Station	0.046	61	0.046	39	0.046	2

表格 5-1表示檢驗安全性質Right_change後的分析表。在一般的局部性分析中的全域系統通常會伴隨全域行動，這使系統狀態空間的增長相當可觀，如圖中的Gas-Station狀態個數為 61，但如果使用ArCats來檢驗安全性質，此時將不需要伴隨多餘的全域行動，Gas-Station的狀態個數也相對的減少，剩下 39 個。使用ArCats的另一項優點在當使用最小化的機制後，依然能夠保留到系統中的死結狀態，這也大幅的減緩組態爆炸的發生，如表中所示，最後Gas-Station剩下兩個狀態。

表格 5-2表示檢驗安全性質Right_service，由於此性質在最後的檢驗中並沒有出現違反狀態，故在做最小化的時候，所有的子系統都有明顯的狀態空間減少趨勢，而最小化後的系統Gas-Station剩下一個狀態。

表格 5-2 比較 CRA 和 ArCats_CRA 在檢驗安全性質 Right_service 時的差異

使用方法 / 子系統	**CRA**_RIGHT_SERVICE		**ARCATS_CRA**_RIGHT_SERVICE			
	Time(sec)	**State#**	**Time(sec)**	**State#**	**Time(sec)**	**Min. State#**
Counter	0.031	13	0.031	13	0.031	13
Station	0.046	35	0.03	35	0.046	25
Client	0.015	16	0.015	16	0.015	16
Gas-Station	0.062	44	0046	28	0.046	1

表格 5-3中顯示了使用ArCats來檢驗安全性質的另一項優點，可以同時檢驗多個安全性質，當我們利用一般的CRA來檢驗Right_change和Right_service時，由於必須分開檢驗，我們將其時間和狀態相加同統計如表中，很明顯的使用ArCats

將會有狀態空間個數較小的優勢，同樣的如果使用ArCats的最小化技術將可以是狀態空間簡化到最小，並且仍然能夠檢驗出是否有違反系統性質的狀態出現。

表格 5-3 比較 CRA 和 ArCats_CRA 同時在檢驗安全性質 Right_service 和 Right_change 時的差異

使用方法 / 子系統	**CRA** RIGHT_CHANGE & IGHT_SERVICE		**ARCATS_CRA** RIGHT_CHANGE & RIGHT_SERVICE			
	Time(sec)	**State#**	**Time(sec)**	**State#**	**Time(sec)**	**Min. State#**
Counter	0.046+0.031	25+13	0.046	25	0.046	25
Station	0.03+0.046	27+35	0.046	43	0.046	43
Client	0.015+0.015	16+16	0.015	16	0.015	16
Gas-Station	0.046+0.062	61+44	0046	39	0.046	2

這裡我們使用的 Gas-Station 並非複雜的系統，當運用在更複雜的系統中將能夠看出使用 ArCats 檢驗安全性質所帶來的種種好處。

5.2 ArCats 和 CRA 在檢驗活化性質的結果分析

使用 ArCats 在檢驗活化性質時相對的比利用一般局部性分析在做檢驗時更具有優勢，原因在於此時輔助檢驗活化性質的不再是使用特殊的狀態，而是以一特殊的全域行動來代替。特殊的狀態在做最小化的過程中將容易被破壞，這使的必須重新去制訂最小化的規則，但如果保留的是全域行動，其將不會因為最小化的過程而被破壞消失。這代表了我們可以即時（on-the fly）的使用最小化的技巧，

來達到檢驗活化性質並同時減少狀態空間的效果。

表格 5-4顯示在ArCats中檢測活化性質Liv_Rec3 過程中， 狀態空間和轉移行動的個數比較，由於ArCats使用 2-pass 的平行合成方式，所以在Pass1 平行合成後將會產生較多的狀態個數，然在經過Pass2 的平行合成後，由於此時才真正將活化行程和子系統做平行合成，故產生的新子系統將會變小（為活化性質保留的action將被平行合成給化減掉）。利用on-the fly Minimization的技巧將會使狀態空間變的更小如表中所示。

表格 5-4 利用 ArCats 檢測 Liv_Rec3 過程比較

使用方法 / 子系統	ARCATS_CRA_LIV_REC_3					
	Pass1 compose		Pass2 compose		Pass2 compose-for min	
	State#	Tran#	State#	Tran#	Min-State#	Min-Tran#
Rec_3	44	172	12	56	6	22

同樣的 ArCats 在檢驗活化性質時，也加入了同時檢驗多個性質的功能，這也使的局部性分析工具有更多樣的選擇。