ПРО ЗАСТОСУВАННЯ ПРАВИЛА КОМПОЗИЦІЇ ПРИ СИНТЕЗІ ФОРМАЛЬНИХ СПЕЦИФІКАЦІЙ

Шкарупило В.В.,
докторант ІПМЕ ім. Г.Є. Пухова НАНУ

Науковий консультант:
д.т.н., О.А. Чемерис

План

- 1. Актуальність.
- 2. Застосування методів перевірки на моделі.
- 3. Специфіка методів перевірки на моделі.
- 4. Акцент роботи.
- 5. Приклад застосування правила композиції.
- 6. Результати застосування.
- 7. Висновки.
- 8. Посилання.

Актуальність

- 1. Системи критичного призначення (СКП, Safety-critical Systems) – системи, збої в роботі яких можуть призвести до катастрофічних наслідків (хімічна промисловість, атомна енергетика тощо).
- 2. У даному контексті оперують поняттям функціональної безпеки, що регламентується стандартом IEC 61508 (функціональна безпека), що полягає у своєчасному виявленні та усуненні потенційно небезпечних умов у роботі електронних систем із програмною складовою, що можуть призвести до негативних наслідків значного масштабу [1].
- Зазначається, що в наш час більше 80% функцій названих систем реалізуються програмно [2].

Актуальність (продовження)

- Зокрема, у Наказі Державної інспекції ядерного регулювання України від 22.07.2015 № 140, із змінами, внесеними згідно з Наказом Державної інспекції ядерного регулювання № 508 від 25.11.2019, дається наступне визначення поняттю функціональної безпеки: «функціональна безпека – властивість системи (компонента) атомної станції, що полягає у здатності виконувати всі потрібні функції, важливі для безпеки, зберігати потрібні властивості та відповідати заданим характеристикам в усіх передбачених проектом режимах й умовах експлуатації» [3].

Застосування методів перевірки на моделі

- 1. Програмна платформа хмарних сервісів Microsoft Azure [4].
- 2. Веб-сервіси Amazon (з 2011 р.) [5];
- 2. Платформа TAS Control Platform, призначена слугувати основою програмної системи керування рухом залізничного транспорту – для досягнення рівня функціональної безпеки SIL 4 (2017 р., Австрія), згідно якого показник інтенсивності відмов системи має знаходитись на рівні 10^-9 (1/год) [6];
- 3. Атомна енергетика Фінляндії (з 2008 р.) [7];
- 4. Операційна система реального часу OpenComRTOS (2010 р.) [8].

Специфіка методів перевірки на моделі

- 1. Характерними ознаками методів перевірки на моделі є наступні: можливість автоматизації їх застосування.
- 2. Судження відносно властивості (властивостей) системи виноситься на основі відповідної моделі – формальної специфікації (ФС).
- 3. Засоби опису ФС різняться як за виразними можливостями, так і за ступенем строгості.

Акцент роботи

- 1. У роботі розглядається формалізм TLA+ темпоральної логіки дій TLA (Temporal Logic of Actions) Л. Лемпорта як засіб створення математично строгих ФС, що, на відміну від альтернативних рішень, дозволяє виразити властивість (властивості) досліджуваної системи єдиною темпоральною формулою [9].
- 2. TLA будується на основі концепції «дії» (Action) - формалізації переходу між станами системи переходів.

Акцент роботи (продовження)

- 1. Компактність і строгість ФС розглядаються в роботі у якості чинників, що спрощують процес сприйняття і аналізу ФС розробником, а також стимулюють зниження впливу людського фактору на результати синтезу ФС.
- 2. У якості засобів розвитку існуючих напрацювань в озвученому контексті розглядаються числення послідовних процесів, що взаємодіють, Ч. Хоара (CSP, Communicating Sequential Processes) [10], «трійки Хоара» і правило композиції [11]:

$$\frac{\{\varphi_0\}e_1\{\varphi_1\},\{\varphi_1\}e_2\{\varphi_2\},\ldots,\{\varphi_{n-1}\}e_n\{\varphi_n\}}{\{\varphi_0\}e_1;e_2;\ldots;e_n\{\varphi_n\}}.$$

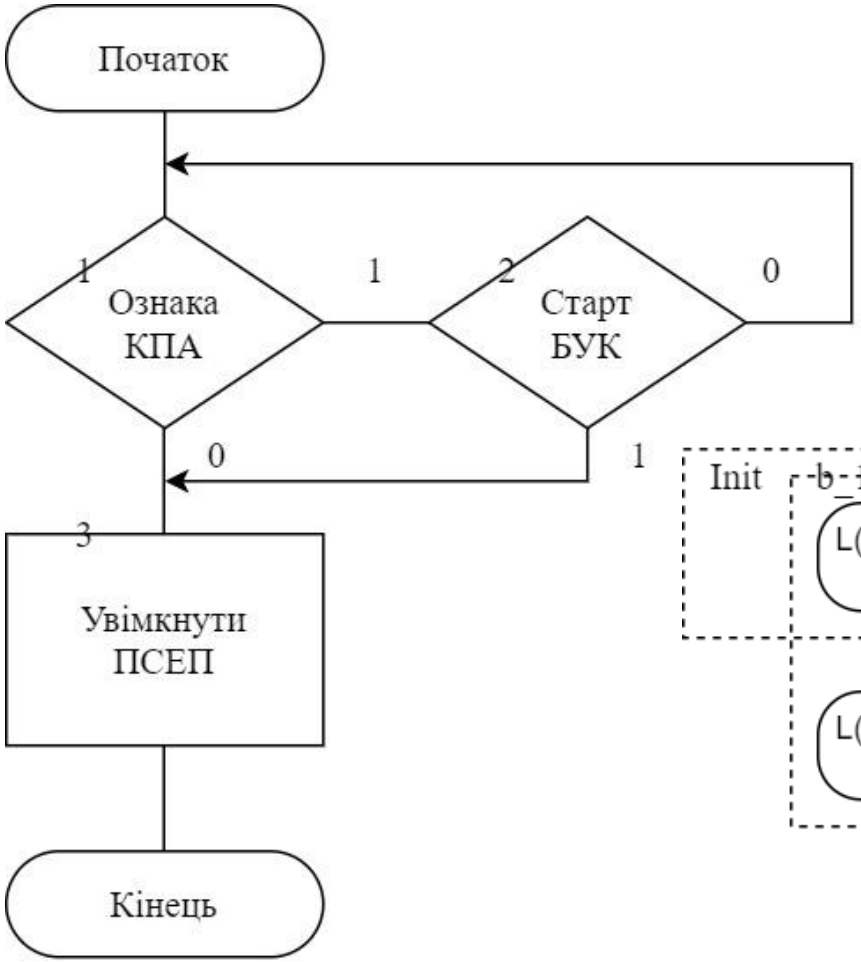
Приклад застосування правила композиції

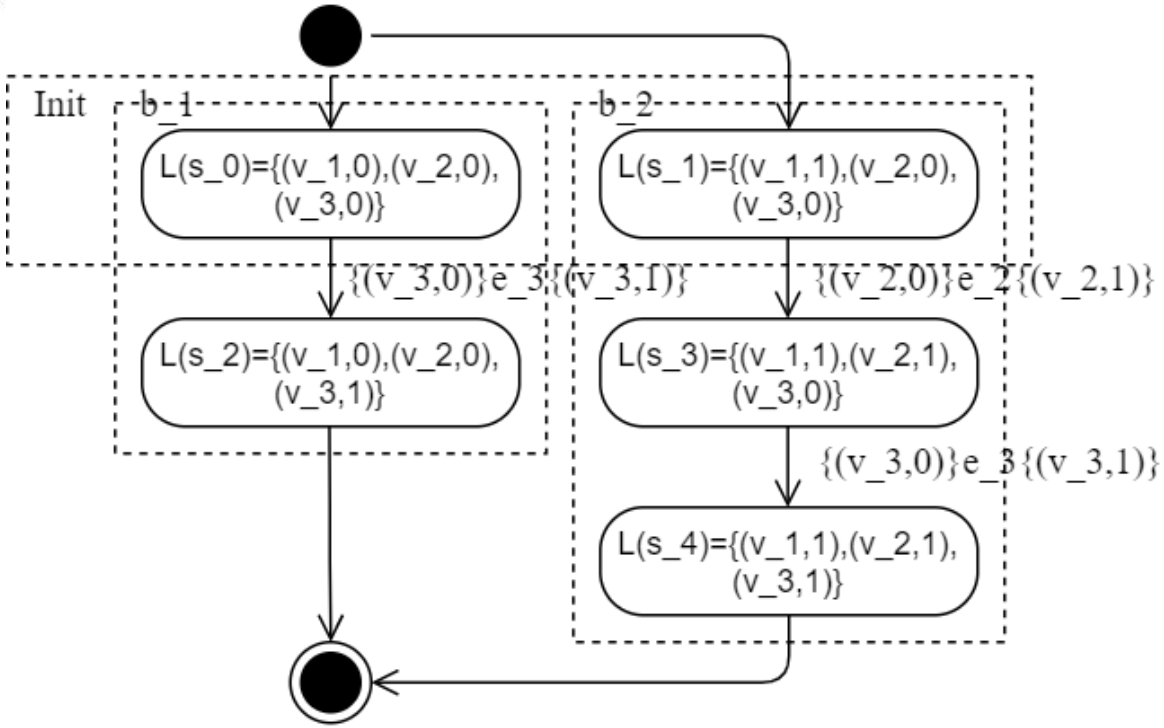
Без застосування	Із застосуванням
// Для n=2^1 ---------- MODULE sec2 ---------- EXTENDS Naturals VARIABLES v1,v2 Invariant == v1 \in (0..2) /\ v2 \in (0..2) Init == v1 = 0 /\ v2 = 0 S1 == v1 = 1 /\ v2 = 0 S2 == v1 = 2 /\ v2 = 0 S3 == v1 = 2 /\ v2 = 1 R1_01 == /\ v1' = IF Init THEN v1+1 ELSE v1 /\ UNCHANGED <<v2>> R1_12 == /\ v1' = IF S1 THEN v1+1 ELSE v1 /\ UNCHANGED <<v2>> R2_23 == /\ v2' = IF S2 THEN v2+1 ELSE v2 /\ UNCHANGED <<v1>> R2_34 == /\ v2' = IF S3 THEN v2+1 ELSE v2 /\ UNCHANGED <<v1>> Next == R1_01 \/ R1_12 \/ R2_23 \/ R2_34 Spec == Init /\[][Next]_<<v1,v2>> =====	// Для n=2^1 ---------- MODULE sec2 ---------- EXTENDS Naturals VARIABLES v1,v2 Invariant == v1 \in (0..2) /\ v2 \in (0..2) Init == v1 = 0 /\ v2 = 0 R1_01 == /\ v1' = IF Init THEN v1+1 ELSE v1 /\ UNCHANGED <<v2>> R1_12 == /\ v1' = IF R1_01 THEN v1+1 ELSE v1 /\ UNCHANGED <<v2>> R2_23 == /\ v2' = IF R1_12 THEN v2+1 ELSE v2 /\ UNCHANGED <<v1>> R2_34 == /\ v2' = IF R2_23 THEN v2+1 ELSE v2 /\ UNCHANGED <<v1>> Next == R1_01 \/ R1_12 \/ R2_23 \/ R2_34 Spec == Init /\[][Next]_<<v1,v2>> =====

Приклад застосування правила композиції

№ з/п	n	Кількість рядків ФС		Різниц, $z - z'$	α
		Без застосування правила, z	Із застосуванням правила, z'		
1	2^1	17	14	3	0,1765
2	2^2	29	22	7	0,2414
3	2^3	53	38	15	0,2830
4	2^4	101	70	31	0,3069
5	2^5	197	134	63	0,3198
6	2^6	389	262	127	0,3265
7	2^7	773	518	255	0,3299
8	2^8	1541	1030	511	0,3316

Приклад СКП


Початок
1
Ознака КПА
1
2
Старт БУК
0
0
1
3
Увімкнути ПСЕП
Кінець


Init
b_1
b_2
L(s_0)={(v_1,0),(v_2,0),(v_3,0)}
L(s_1)={(v_1,1),(v_2,0),(v_3,0)}
{(v_3,0)}e_3{(v_3,1)}
{(v_2,0)}e_2{(v_2,1)}
L(s_2)={(v_1,0),(v_2,0),(v_3,1)}
L(s_3)={(v_1,1),(v_2,1),(v_3,0)}
{(v_3,0)}e_3{(v_3,1)}
L(s_4)={(v_1,1),(v_2,1),(v_3,1)}

Відповідна специфікація

Із застосуванням правила

```
-------- MODULE spec --------
EXTENDS Naturals
VARIABLES v1,v2,v3
\* 5 states, depth = 3
\* obtained manually
Invariant == v1 \in {0,1} /\ v2 \in {0,1} /\ v3 \in {0,1}
Init == v1 \in {0,1} /\ v2 = 0 /\ v3=0
R02 == /\ v3' = IF v1=0 /\ v3=0 THEN 1-v3 ELSE v3
      /\ UNCHANGED <<v1,v2>>
R13 == /\ v2' = IF v1=1 /\ v2=0 THEN 1-v2 ELSE v2
      /\ UNCHANGED <<v1>>
R34 == /\ v3' = IF R13 THEN 1-v3 ELSE v3
      /\ UNCHANGED <<v1>>

Next == R02 \/ (R13 /\ R34)

Spec == Init/\[][Next]_<<v1,v2,v3>>
========
```

Висновки

- Результати проведених експериментальних досліджень показали, що, на прикладі досліджуваного тестового набору, застосування правила композиції дозволило зменшити розмір результуючої ФС на значення від 18 до 33 %. Такий ефект охарактеризовано як вагомий. При цьому показником розміру ФС виступила кількість рядків.
- Отже, застосування правила композиції на основі трійок Хоара є дієвим засобом компактизації формальної специфікації системи.

Посилання

- 1. IEC 61508 Edition 2.0. Functional safety of electrical/electronic/programmable electronic safety-related systems. [Approved: April 2010]. URL: https://www.iec.ch/functionalsafety/standards/page2.htm. (Accessed: 07.02.2020).
- 2. Омельчук Л. Л. Формальні методи специфікації програм: навч. посібник. К. : УкрІНТЕІ, 2010. 78 с.
- 3. Про затвердження Вимог з ядерної та радіаційної безпеки до інформаційних та керуючих систем, важливих для безпеки атомних станцій: наказ Державної інспекції ядерного регулювання від 22.07.2015 № 140, із змінами, внесеними згідно з Наказом Державної інспекції ядерного регулювання № 508 від 25.11.2019. URL: https://zakon.rada.gov.ua/laws/term/34229 (дата звернення: 26.03.2020).
- 4. Kuppe M. A., Lamport L., Ricketts D. The TLA+ Toolbox. Formal Integrated Development Environment, F-IDE 2019 : 5th Workshop (Porto, Portugal, October 7, 2019). EPTCS 310, 2019. P. 50-62. DOI: http://doi.org/10.4204/EPTCS.310.6

Посилання (продовження)

- 5. C. Newcombe, T. Rath, F. Zhang, B. Munteanu, M. Brooker, and M. Deardeuff, “How Amazon web services uses formal methods,” Communications of the ACM, vol. 58, no. 4, pp. 66-73, 2015.
- 6. Resch S., Paulitsch M. Using TLA+ in the Development of a Safety-Critical Fault-Tolerant Middleware. Software Reliability Engineering Workshops : Proc. 2017 IEEE International Symposium on Software Reliability Engineering Workshops (Toulouse, France, 23-26 October 2017). P. 146-152. doi: https://doi.org/10.1109/ISSREW.2017.43
- 7. A. Pakonen, T. Tahvonen, M. Hartikainen, and P. Mikko, “Practical applications of model checking in the Finnish nuclear industry,” in Proc. 10th International Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies, NPIC & HMIT 2017, San Francisco, CA, USA, 2017, pp. 1342–1352.
- 8. Verhulst E., Boute R. T., Faria J. M. S., Sputh B. H. C., Mezhuyev V. Formal Development of a Network-Centric RTOS: Software Engineering for Reliable Embedded Systems. Springer Publishing Company, Inc., 2011. 236 p.

Посилання (продовження)

- 9. Lamport L. Specifying systems: The TLA+ language and tools for hardware and software engineers. Boston : Addison-Wesley, 2002. 382 p.
- 10. Hoare C. A. R. Communicating sequential processes. Communications of the ACM. 1978. Vol. 21, No. 8. P. 666-677.
- 11. Hoare C. A. R. An axiomatic basis for computer programming. Communications of the ACM. 1969. Vol. 12, No. 10. P. 576-583.