Zastosowanie bisymulacji do weryfikowania własności Non-Interference

XVI FIT, Karpacz 2002

Wojciech Tomanik, Wiktor Zychla

Uniwersytet Wrocławski

Instytut Informatyki

14 grudnia 2002

’Bezpieczeństwo’ - dla nas

- Pojęcie poufności
  - program czyta dane z dysku, następnie wysyła je w świat
  - program próbuje czytać dane, których nie powinien czytać
- Pojęcie przepływu informacji
  - stan systemu zmienia się w czasie działania programu
- Pojęcie interakcji
  - wiele składników systemu działa jednocześnie, może więc wchodzić w przeróżne interakcje
- Pojęcie niedeterminizmu
  - współczesne systemy działają niedeterministycznie - kolejność wykonywania akcji procesów równoległych jest nieokreślona

Non-Interference

- W najbardziej ogólnym sformułowaniu, Non-Interference oznacza brak przepływu informacji między równocześnie działającymi agentami
- W naturalny sposób można to pojęcie wykorzystać do sprawdzania czy system jest bezpieczny: system jest bezpieczny, gdy nie ma w nim niepożądanych przepływów informacji

Non-Interference, świat H-L

Załóżmy, że wszystkie akcje klasyfikujemy jako jawne (bezpieczne, publiczne, L) lub tajne (niebezpieczne, prywatne, H).

- Dwa poziomy dostępu wystarczą do analizy bardziej złożonych scenariuszy.
- Klasyfikacja akcji jest punktem wyjścia do dalszej analizy systemu. Non-Interference nie narzuca tutaj jednak żadnych ograniczeń.
- Użytkownik L nie powinien w trakcie wykonywania swojego programu mieć możliwości bezpośrednio ani pośrednio wchodzić w posiadanie informacji nie przeznaczonych dla niego

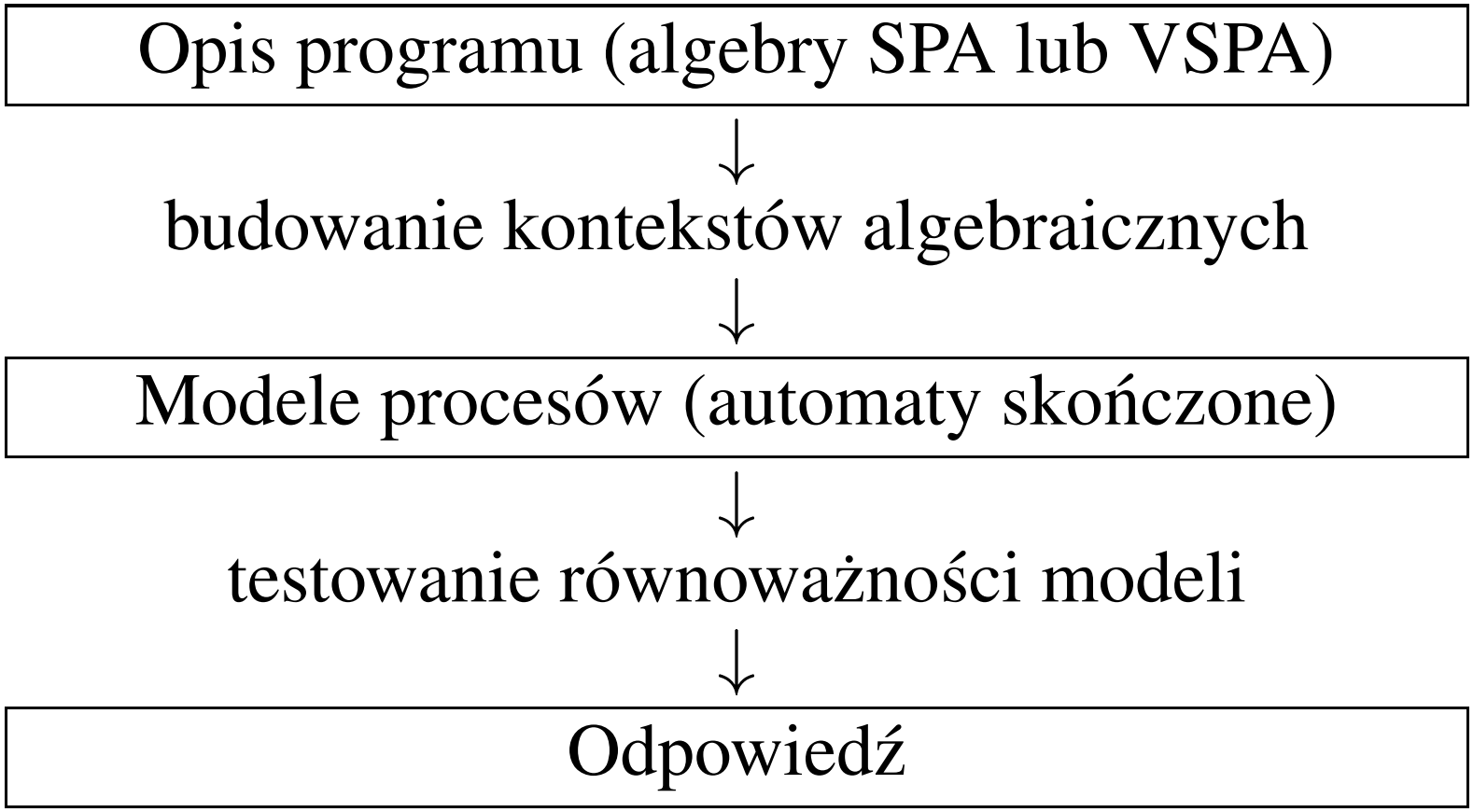
Podobnie użytkowników systemu dzielimy na użytkowników L i H.

Non-Interference, definicja

- System nazwiemy bezpiecznym w sensie Non-Interference, jeśli nie ma żadnej interferencji między akcjami jawnymi i tajnymi, to znaczy że obserwując wykonanie się swojego programu, użytkownik L nie potrafi wydedukować czy w systemie wykonują się jakiekolwiek akcje tajne

Okazuje się, że w pewnych przypadkach jest to polisa zbyt silna. Mimo to, Non-Interference potrafi wykrywać wiele jawnych oraz niejawnych kanałów możliwego wycieku informacji.

System analizy NI



Rachunek SPA

- Rachunek SPA jest rozwinięciem rachunku CCS, zaproponowanego przez Milnera.
- Składnia rachunku:

$$E ::= \underline{0} \mid \mu.E \mid E + E \mid E|E \mid E\backslash\backslash L \mid E[f] \mid Z$$

- Act_H - akcje użytkownika uprzywilejowanego
- Act_L - akcje użytkownika nieuprzywilejowanego
- wszystkie akcje:
  $Act = Act_H \cup Act_L \cup \{\tau\}$
- akcje wejściowe i akcje wyjściowe: a, $\overline{a}$

Kontesty algebraiczne

- Różne własności NI testujemy w tzw. kontekstach agebraicznych.
- Załóżmy, że chcemy sprawdzić proces P.
- Umieszczamy P w różnych kontekstach, następnie generujemy odpowiednie LTS.
- Testujemy równoważność otrzymanych LTS.
- Dobierając różne konteksty dla procesu P badamy różne własności.

Równoważność automatów

Testowanie równoważności automatów będących modelami różnych procesów jest podstawowym narzędziem NNI. Wiele jest definicji równoważności automatów.

- równoważność śladowa, $\approx_T$
- równoważność bisymulacyjna, $\approx_B$

Rodzaje nieinterferencji

- NNI

$$(P\backslash\backslash_I Act_H)/Act_H \approx_T P/Act_H$$

- SNNI

$$P\backslash\backslash Act_H \approx_T P/Act_H$$

- BNNI

$$(P\backslash\backslash_I Act_H)/Act_H \approx_B P/Act_H$$

- BSNNI

$$P\backslash\backslash Act_H \approx_B P/Act_H$$

Bisymulacja

Dla danego automatu skończonego $S = (Q, Act, T, q_0)$, binarna relacja $\rho \subseteq Q \times Q$ jest bisymulacją gdy:

$$\forall (p_1, p_2) \in \rho, \forall a \in Act$$
$$\forall r_1.(p_1 \xrightarrow{a} r_1 \Rightarrow \exists r_2.(p_2 \xrightarrow{a} r_2 \wedge (r_1, r_2) \in \rho)) \wedge$$
$$\forall r_2.(p_2 \xrightarrow{a} r_2 \Rightarrow \exists r_1.(p_1 \xrightarrow{a} r_1 \wedge (r_1, r_2) \in \rho))$$

Algorytm polega na skonstruowaniu klas $\{B_1, \ldots, B_n\}$ które reprezentują równoważne stany w automacie S, rozpoczynając od najbardziej uniwersalnej relacji ρ_U.

Rozróżnienie $\approx_T$ i $\approx_B$

Mamy dwa automaty dane przez konteksty algebraiczne NI.
W jaki sposób system odróżnia $\approx_T$ i $\approx_B$?

- W przypadku obu relacji buduje się tranzytywne domknięcie relacji τ
- W przypadku $\approx_T$ automaty są determinizowane (klasyczny algorytm determinizacji automatu skończonego)

Teraz oba automaty są przekazywane do jądra algorytmu, które testuje ich równoważność. Dzięki jednorodnemu podejściu łatwiej jest rozbudowywać system o dodatkowe wersje NI oraz inne rodzaje równoważności.

NIC - Non-Interference Checker

System stworzony od zera, potrafi **wszystko** o czym mówiliśmy

- Budowanie modeli w kontekstach algebraicznych - Java (Wojtek)
- Fazy wstępne + test Paige-Tarjan - C# (Wiktor)

Złożoność:

- Budowanie modeli
  - Operator | powoduje wykładniczy wzrost ilości stanów (*state explosion*).
- Fazy wstępne + test bisymulacji
  - Determinizacja automatu skończonego może powodować wykładniczy wzrost ilości stanów. W praktyce okazuje się, że często powstaje automat o porównywalnej ilości stanów.
  - Sam algorytm PT ma złożoność $O(m \log n)$

Przykład - Monitor dostępu

$$
\begin{aligned}
AccessMonitor1 &= (Monitor \mid Object(1,0) \mid Object(0,0)) \setminus (r,w) \\
Object(x,y) &= \overline{r}(x,y) + Object(x,y) + w(x,z).Object(x,z) \\
Monitor &= accessR(l,x). \\
&\quad (\textbf{if } x \leq l \textbf{ then} \\
&\quad\ r(x,y).\overline{val}(l,y).Monitor \\
&\quad \textbf{else} \\
&\quad\ \overline{val}(l,err).Monitor) + \\
&\quad accessW(l,x).write(l,z). \\
&\quad (\textbf{if } x \geq l \textbf{ then} \\
&\quad\ \overline{w}(x,z).Monitor \\
&\quad \textbf{else} \\
&\quad\ Monitor)
\end{aligned}
$$

Monitor dostępu, pułapka

$$
\begin{aligned}
AccessMonitor2 &= (Monitor \mid Object(1,0) \mid Object(0,0)) \setminus (r,w) \\
Object(x,y) &= \overline{r}(x,y) + Object(x,y) + w(x,z).Object(x,z) \\
Monitor &= accessR(l,x). \\
&\quad (\textbf{if } x \leq l \textbf{ then} \\
&\quad\ r(x,y).\overline{val}(l,y).Monitor \\
&\quad \textbf{else} \\
&\quad\ \overline{val}(l,err).Monitor) + \\
&\quad accessW(l,x).write(l,z).\overline{w}(x,z).Monitor
\end{aligned}
$$

Użytkownik H może zapisać wartość do obiektu L, łamiąc reguły polisy bezpieczeństwa.

Monitor dostępu, analiza

Wyniki analizy obu Monitorów dostępu:

- AccessMonitor1 jest NNI
- AccessMonitor2 nie jest NNI
- Na przeciętnym PC (500Mhz) test w obu przypadkach trwa krócej niż 1 sekundę

Za pomocą pozostałych wariantów NI można analizować o wiele bardziej subtelne sposoby przekazywania informacji między światami L i H (tzw. *covert channels* czyli kanały ukryte).

Perspektywy

Kierunki możliwych badań:

- Inne warianty Non-Interference, relacje między nimi
- Inne rodzaje równoważności, relacje między nimi
- Modele symboliczne
- Lepsze algorytmy testowania bisymulacji

Koniec

Dziękujemy za uwagę