Formale Grundlagen der Informatik 3

Concurrency, Fairness,

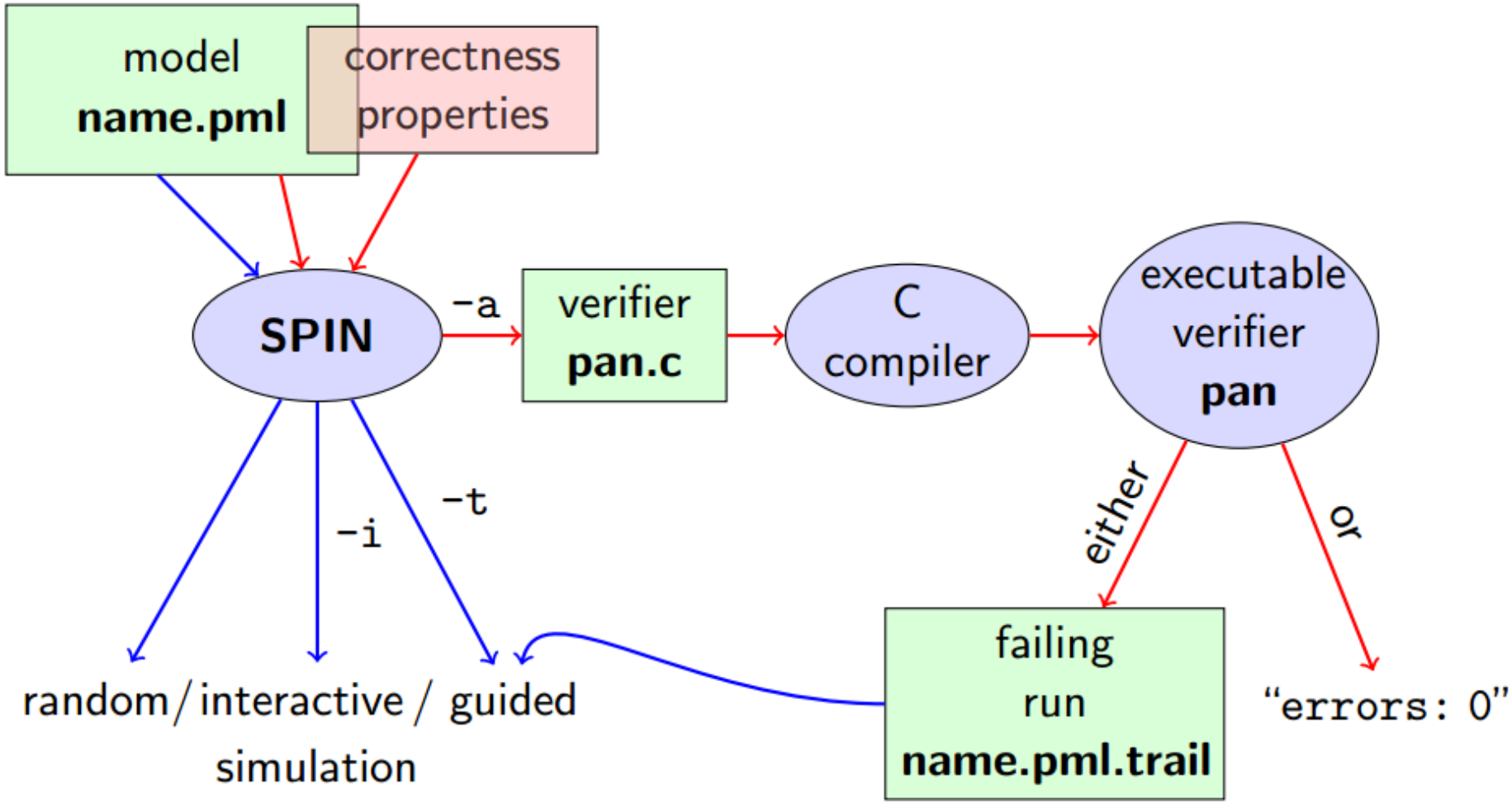
SPIN Wiederholung

TECHNISCHE UNIVERSITÄT DARMSTADT

Prof. Stefan Katzenbeisser
Security Engineering Group
Technische Universität Darmstadt

skatzenbeisser@acm.org
http://www.seceng.informatik.tu-darmstadt.de

SPIN (Wiederholung)
Workflow: Übersicht



SPIN (Wiederholung) Korrektheitseigenschaften

TECHNISCHE UNIVERSITÄT DARMSTADT

Korrektheitseigenschaften können innerhalb oder außerhalb des PROMELA Modells angegeben werden

Innerhalb des Modells:

- Assertion Statements
- „Meta Labels"

Außerhalb des Modells:

- Never Claims
- LTL-Formel

Ziele:

- Deadlocks, Race Conditions, Verletzung von Aussagen erkennen
- Safety / Liveness Eigenschaften sicherstellen

SECENG SECURITY ENGINEERING

SPIN
Fairness

TECHNISCHE UNIVERSITÄT DARMSTADT

Wird das folgende Modell in jedem Durchlauf terminieren?

```
byte n = 0;
bool flag = false;
active proctype P(){
do
  :: flag -> break;
  :: else -> n = 5 - n
od
}
active proctype Q(){
flag = true
}
```

➔ Terminiert nur, wenn das Scheduling „fair" ist

Weak Fairness: Ein Ablauf wird **fair** genannt, wenn darin jeder Prozess der unendlich oft „aktiviert" ist auch unendlich oft Rechenzeit bekommt.

SECENG SECURITY ENGINEERING

SPIN
Concurrency / Interleaving

TECHNISCHE UNIVERSITÄT DARMSTADT

- SPIN simuliert PROMELA Modelle nicht-deterministisch
  - zufällige Wahl bei mehreren erfüllten Guards
  - zufällige Wahl des nächsten Prozesses in der Ausführungsfolge
- Bei gemeinsamen Variablen kann unvorhergesehenes Verhalten entstehen!
- Frage: Wie viele Ausgaben sind im Beispiel möglich?

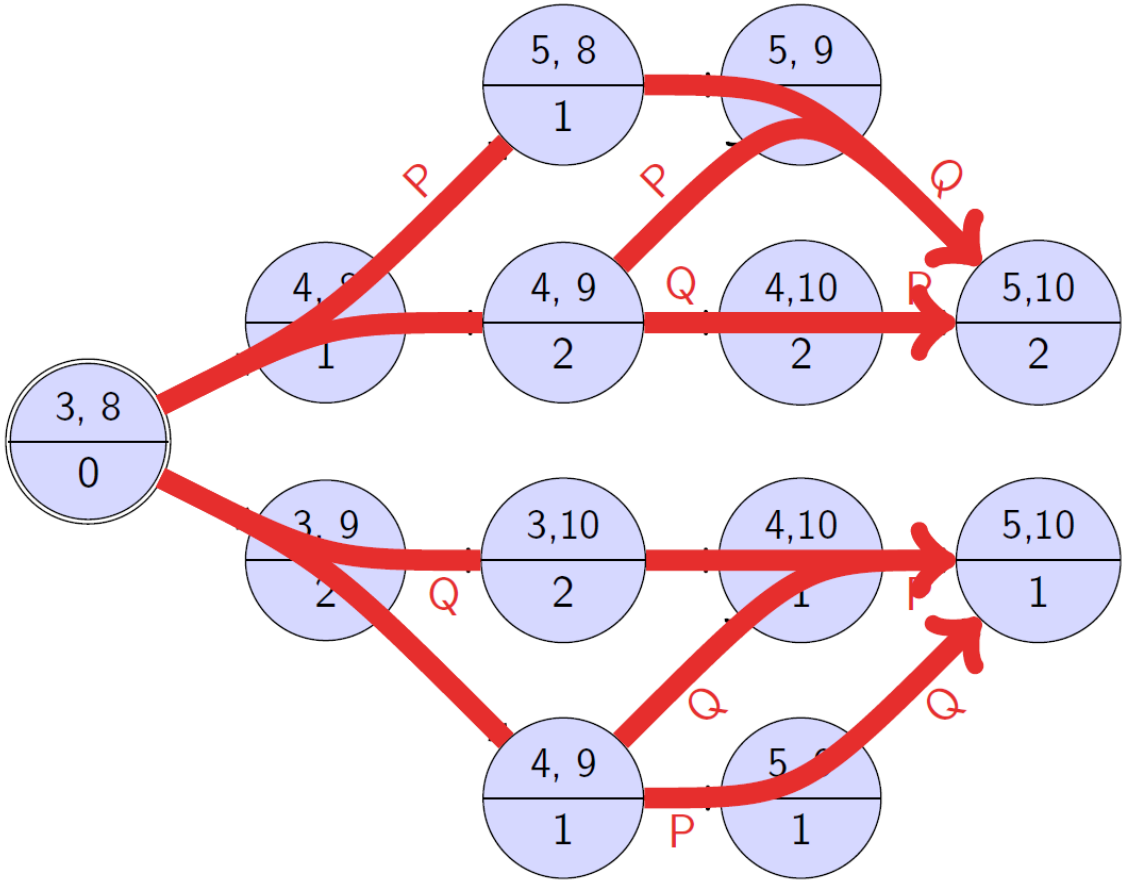
```
1       byte n = 0;
2
3       active proctype P(){
4         n = 1;
5         printf("Process P, n = %d\n", n)
6       }
7
8       active proctype Q(){
9         n = 2;
10        printf("Process Q, n = %d\n", n)
11      }
```


SECENG Security Engineering

SPIN
Concurrency / Interleaving (2)

TECHNISCHE UNIVERSITÄT DARMSTADT

```
1  byte n = 0;
2
3  active
   proctype P(){
4   n = 1;
5   printf("
    Process P,
    n = %d\n", n)
6   }
7
8  active
   proctype Q(){
9   n = 2;
10 printf("
    Process Q,
    n = %d\n", n)
11 }
```



SECENG SECURITY ENGINEERING

SPIN Beispiel – Needham-Schroeder Schlüsseltausch, Einführung (1)

TECHNISCHE UNIVERSITÄT DARMSTADT

Ziel des Protokolls:

Kryptographische Schlüssel zwischen zwei Parteien aushandeln

Ablauf des Protokolls:

X -> Y: $\{X, NX\}_{kY}$

Y -> X: $\{NX, NY\}_{kX}$

X -> Y: $\{NY\}_{kY}$

$\{msg\}_{kp}$: msg wird mit kp verschlüsselt (public key)

SECENG SECURITY ENGINEERING

SPIN Beispiel – Needham-Schroeder Schlüsseltausch, Einführung (2)

TECHNISCHE UNIVERSITÄT DARMSTADT

- zwei Teilnehmer (Alice, Bob) mit public keys (kA, kB)
- nonceA und nonceB zur Vereinfachung statisch
  (aber: A kennt nur nonceA; B kennt nur nonceB!)
- verschlüsselte Nachrichten repräsentiert durch Crypt-Struktur;
  Entschlüsselung durch Vergleich der key Einträge

```
mtype = {ok, err, msg1, msg2, msg3, keyA, keyB, agentA,
  agentB, nonceA, nonceB };
typedef Crypt { mtype key, content1, content2};
chan network = [0] of {mtype, /* msg# */
                       mtype, /* receiver */
                       Crypt};
```


SECENG SECURITY ENGINEERING

SPIN Beispiel – Needham-Schroeder Schlüsseltausch, Aufgaben

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Erstelle ein Model für Alice und Bob
2. Verifizieren Sie, dass beide Modelle komplett ausgeführt werden
3. Einführung eines Angreifers (Dolev-Yao)
   1. Angreifer kann Nachrichten auslesen, Replays oder das Protokoll neu starten
   2. Prüfen Sie, ob die Bedingung unter 2 noch gilt
4. In 3 ist der Angreifer komplett extern; wir nehmen nun an, dass dieser ein legitimer Nutzer des Systems ist
   1. Passen Sie den Code so an, dass nichtdeterministisch zwischen den Teilnehmern gewählt wird
5. Spezifikation von Security Garantien (in LTL)
   1. Wenn A und B ihren Ablauf komplettieren, sind A und B Kommunikationspartner
   2. Wenn A am Ende ankommt und glaubt mit B zu sprechen, kennt AT A's nonce nicht
   3. Wie 2 nur für B

SECENG SECURITY ENGINEERING

SPIN Beispiel – Needham-Schroeder Schlüsseltausch, Aufgaben

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Erstelle ein Model für Alice und Bob
2. Verifizieren Sie, dass beide Modelle komplett ausgeführt werden
3. Einführung eines Angreifers (Dolev-Yao)
   1. Angreifer kann Nachrichten auslesen, Replays oder das Protokoll neu starten
   2. Prüfen Sie, ob die Bedingung unter 2 noch gilt
4. In 3 ist der Angreifer komplett extern; wir nehmen nun an, dass dieser ein legitimer Nutzer des Systems ist
   1. Passen Sie den Code so an, dass nichtdeterministisch zwischen den Teilnehmern gewählt wird
5. Spezifikation von Security Garantien (in LTL)
   1. Wenn A und B ihren Ablauf komplettieren, sind A und B Kommunikationspartner
   2. Wenn A am Ende ankommt und glaubt mit B zu sprechen, kennt AT A's nonce nicht
   3. Wie 2 nur für B

SECENG SECURITY ENGINEERING

SPIN Beispiel – Needham-Schroeder
Aufgabe 1 – Modellierung

TECHNISCHE UNIVERSITÄT DARMSTADT

```
active proctype Alice() {
mtype pkey; /* the other agent's public
    key */
mtype pnonce; /* from other agent */
Crypt messageAB; /* our encrypted
    message to other party */
Crypt data; /* received encrypted
    message */

partnerA = agentB;
pkey     = keyB;
/* Prepare the first message */
messageAB.key = pkey;
messageAB.content1 = agentA;
messageAB.content2 = nonceA;
/* Send the first message */
network ! msg1 (partnerA, messageAB);
…
}
```

```
active proctype Bob() {
…
partnerB = agentA;
pkey = keyA;
/* Wait for a message. */
network ? (msg1, agentB, data);
/* Check the key */
(data.key == keyB) && (data.content1 ==
    partnerB);
pnonce = data.content2;
/* Prepare second message */
messageBA.key = pkey;
messageBA.content1 = pnonce;
messageBA.content2 = nonceB;
/* Send the second message to A */
network ! msg2 (partnerB, messageBA);
…
}
```


SECENG SECURITY ENGINEERING

SPIN Beispiel – Needham-Schroeder Schlüsseltausch, Aufgaben

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Erstelle ein Model für Alice und Bob
2. Verifizieren Sie, dass beide Modelle komplett ausgeführt werden
3. Einführung eines Angreifers (Dolev-Yao)
    1. Angreifer kann Nachrichten auslesen, Replays oder das Protokoll neu starten
    2. Prüfen Sie, ob die Bedingung unter 2 noch gilt
4. In 3 ist der Angreifer komplett extern; wir nehmen nun an, dass dieser ein legitimer Nutzer des Systems ist
    1. Passen Sie den Code so an, dass nichtdeterministisch zwischen den Teilnehmern gewählt wird
5. Spezifikation von Security Garantien (in LTL)
    1. Wenn A und B ihren Ablauf komplettieren, sind A und B Kommunikationspartner
    2. Wenn A am Ende ankommt und glaubt mit B zu sprechen, kennt AT A's nonce nicht
    3. Wie 2 nur für B

SECENG SECURITY ENGINEERING

SPIN Beispiel – Needham-Schroeder
Aufgabe 2 – Verifizieren, dass A & B bis Ende laufen

TECHNISCHE UNIVERSITÄT DARMSTADT

```
active proctype Alice() {
…
statusA = ok;
}
```

```
active proctype Bob() {
…
statusB = ok;
}
```

```
ltl BOTH_ARE_OK {<> (statusA == ok && statusB == ok)}
```


SECENG SECURITY ENGINEERING

SPIN Beispiel – Needham-Schroeder Schlüsseltausch, Aufgaben

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Erstelle ein Model für Alice und Bob
2. Verifizieren Sie, dass beide Modelle komplett ausgeführt werden
3. Einführung eines Angreifers (Dolev-Yao)
   1. Angreifer kann Nachrichten auslesen, Replays oder das Protokoll neu starten
   2. Prüfen Sie, ob die Bedingung unter 2 noch gilt
4. In 3 ist der Angreifer komplett extern; wir nehmen nun an, dass dieser ein legitimer Nutzer des Systems ist
   1. Passen Sie den Code so an, dass nichtdeterministisch zwischen den Teilnehmern gewählt wird
5. Spezifikation von Security Garantien (in LTL)
   1. Wenn A und B ihren Ablauf komplettieren, sind A und B Kommunikationspartner
   2. Wenn A am Ende ankommt und glaubt mit B zu sprechen, kennt AT A's nonce nicht
   3. Wie 2 nur für B

SECENG SECURITY ENGINEERING

SPIN Beispiel – Needham-Schroeder
Aufgabe 3 – Angreifer (Exemplarisch)

TECHNISCHE UNIVERSITÄT DARMSTADT

```
active proctype Intruder() {
…
do
  :: /* Replay or send a message */
  …
  if /* replay intercepted message or assemble it */
    :: data.key    = intercepted.key;
       data.content1  = intercepted.content1;
       data.content2  = intercepted.content2;
    :: if /* assemble content1 */
       :: data.content1 = agentA;
       :: data.content1 = agentB;
       :: data.content1 = agentI;
       :: data.content1 = nonceI;
    fi;
       data.content2 = nonceI;
    fi;
    if /* assemble key */
       :: data.key = keyA;
       :: data.key = keyB;
       :: data.key = keyI;
    fi;
    network ! msg (recpt,
       data);
od
…
}
```

```
ltl BOTH_ARE_OK {<> (statusA == ok && statusB == ok)}
```


SECENG SECURITY ENGINEERING

SPIN Beispiel – Needham-Schroeder Schlüsseltausch, Aufgaben

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Erstelle ein Model für Alice und Bob
2. Verifizieren Sie, dass beide Modelle komplett ausgeführt werden
3. Einführung eines Angreifers (Dolev-Yao)
   1. Angreifer kann Nachrichten auslesen, Replays oder das Protokoll neu starten
   2. Prüfen Sie, ob die Bedingung unter 2 noch gilt
4. In 3 ist der Angreifer komplett extern; wir nehmen nun an, dass dieser ein legitimer Nutzer des Systems ist
   1. Passen Sie den Code so an, dass nichtdeterministisch zwischen den Teilnehmern gewählt wird
5. Spezifikation von Security Garantien (in LTL)
   1. Wenn A und B ihren Ablauf komplettieren, sind A und B Kommunikationspartner
   2. Wenn A am Ende ankommt und glaubt mit B zu sprechen, kennt AT A's nonce nicht
   3. Wie 2 nur für B

SECENG SECURITY ENGINEERING

SPIN Beispiel – Needham-Schroeder
Aufgabe 4 – Erweitern des Angreifers

TECHNISCHE UNIVERSITÄT DARMSTADT

```
active proctype Alice() {
…
if
  :: partnerA = agentI;
    pkey     = keyI;
  :: partnerA = agentB;
    pkey     = keyB;
fi;
…
}
```

```
active proctype Intruder() {
…
if
  :: (data.key == keyI) && (data.content2 ==
    nonceA) -> knows_nonceA = true;
  :: (data.key == keyI) && (data.content1 ==
    nonceB) -> knows_nonceB = true;
fi;
…
if /* assemble content2 */
  :: msg == msg3 -> data.content2 = 0;
  :: else -> if
    :: knows_nonceA -> data.content2 = nonceA;
    :: knows_nonceB -> data.content2 = nonceB;
    :: data.content2 = nonceI;
  fi;
fi;
…
}
```


SECENG SECURITY ENGINEERING

SPIN Beispiel – Needham-Schroeder Schlüsseltausch, Aufgaben

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Erstelle ein Model für Alice und Bob
2. Verifizieren Sie, dass beide Modelle komplett ausgeführt werden
3. Einführung eines Angreifers (Dolev-Yao)
   1. Angreifer kann Nachrichten auslesen, Replays oder das Protokoll neu starten
   2. Prüfen Sie, ob die Bedingung unter 2 noch gilt
4. In 3 ist der Angreifer komplett extern; wir nehmen nun an, dass dieser ein legitimer Nutzer des Systems ist
   1. Passen Sie den Code so an, dass nichtdeterministisch zwischen den Teilnehmern gewählt wird
5. Spezifikation von Security Garantien (in LTL)
   1. Wenn A und B ihren Ablauf komplettieren, sind A und B Kommunikationspartner
   2. Wenn A am Ende ankommt und glaubt mit B zu sprechen, kennt AT A's nonce nicht
   3. Wie 2 nur für B

SECENG SECURITY ENGINEERING

SPIN Beispiel – Needham-Schroeder
Aufgabe 5 – Security Garantien

TECHNISCHE UNIVERSITÄT DARMSTADT

Wenn A und B ihren Ablauf komplettieren, sind A und B Kommunikationspartner:

```
ltl PROP_AB { [](statusA == ok && statusB == ok) ->
   (partnerA == agentB && partnerB == agentA)) }
```

Wenn A am Ende ankommt und glaubt mit B zu sprechen, kennt AT A's nonce nicht:

```
ltl PROP_A { [](statusA == ok && partnerA == agentB) ->
   !knows_nonceA) }
```

Wie vorheriges, aber für B:

```
ltl PROP_B { [](statusB == ok && partnerB == agentA) ->
   !knows_nonceB) }
```


SECENG SECURITY ENGINEERING