Geteiltes Wissen und Handlungskoordination

Vera Derkacheva
Folien: H. Leiß

LMU München, CIS
Seminar Geteiltes Wissen und Spieltheoretische Semantik
SS 2011

3.6.2011

Beispiel: koordinierter Angriff

Problem

(„coordinated attack“) Zwei Generäle können einen gemeinsamen Feind nur besiegen, wenn sie ihn gleichzeitig angreifen. Also greift keiner der Generäle an, bevor er nicht weiß, daß der andere auch angreift, d.h. bevor er nicht vom anderen eine Nachricht erhält, daß dieser angreift. Eine Nachrichtenübertragung dauert eine Stunde.

Können die Generale durch Nachrichten einen gemeinsamen Angriff absprechen?

Definitionen

Sprachen

- $\mathcal{L}_n(\Phi)$ Sprache mit Formeln

$$\varphi, \psi ::= p(\in \Phi) \mid \top \mid \neg\varphi \mid (\varphi \wedge \psi) \mid K_1\varphi \mid \ldots \mid K_n\varphi$$

- $\mathcal{L}_n^C(\Phi)$ Sprache mit Formeln

$$\varphi, \psi ::= p(\in \Phi) \mid \top \mid \neg\varphi \mid (\varphi \wedge \psi) \mid K_1\varphi \mid \ldots \mid K_n\varphi \mid C\varphi$$

Modellklasse

- $\mathcal{M}_n^{rst}$: Klasse aller Modelle $M = (S, \mathcal{K}_1, \ldots, \mathcal{K}_n, \pi)$ mit Äquivalenzrelationen $\mathcal{K}_i \subseteq S \times S$ und $\pi : S \to \Phi \to \mathbb{B}$.

Gültigkeit:

1. *in M gilt φ (lokal) in der Welt s*, $(M, s) \models \varphi$, definiert mit Hilfe von

$$(M, s) \models K_i\varphi : \Longleftrightarrow \text{ für alle } t \text{ mit } (s, t) \in \mathcal{K}_i \text{ ist } (M, t) \models \varphi,$$

2. *in M gilt φ (global) in allen Welten*,

$$M \models \varphi \ : \Longleftrightarrow \ \text{für alle } s \in S \text{ ist } (M, s) \models \varphi,$$

3. *in $\mathcal{M}$ gilt φ allgemein*,

$$\mathcal{M} \models \varphi \ : \Longleftrightarrow \ \text{für alle } M \in \mathcal{M} \text{ ist } M \models \varphi.$$

Kripke-Struktur $M_{\mathcal{I}}$ eines Systems $\mathcal{I}$

Definition

Ein **System** $\mathcal{R}$ über der Menge

$$G = L_0 \times L_1 \times \ldots \times L_n$$

ist eine Menge von **Abläufen** $r \in (\mathbb{N} \to G)$. Ein **interpretiertes System** $\mathcal{I} = (\mathcal{R}, \pi)$ über G ist ein System $\mathcal{R}$ über G mit einer Interpretation $\pi : G \times \Phi \to \{\textbf{true}, \textbf{false}\}$ der Aussagen von Φ.

Definition

Die globalen Zustände $s = (s_0, \ldots, s_n)$ und $s' = (s'_0, \ldots, s'_n) \in G$ sind **für Agent** i **ununterscheidbar**, $s \sim_i s'$, wenn $s_i = s'_i$.

Definition

Die Kripke-Struktur

$$\mathbf{M}_{\mathcal{I}} = (S, \pi, K_1, \ldots, K_n)$$

zu $\mathcal{I} = (\mathcal{R}, \pi)$ besteht aus der Menge

$$S := \mathcal{R} \times \mathbb{N}$$

aller **Punkte** (r, m) von $\mathcal{I}$ und den Erreichbarkeitsrelationen $K_i := \sim_i \subseteq S \times S$ mit

$$(r, m) \sim_i (r', m') : \Longleftrightarrow \; r(m)_i = r'(m')_i.$$

Hierbei ist $r(m) = (r(m)_0, \ldots, r(m)_n) \in G$.

Jedem Agenten $i = 1, \ldots, n$ und der Umgebung $e = 0$ sei eine Menge ACT_i von **Aktionen** zugeordnet, die er/es ausführen kann.

Definition

Eine **Übergangsfunktion** auf G ist eine Abbildung $\tau : G \to G$.
Der **gemeinsamen Aktion**

$$a = (a_0, \ldots, a_n) \in ACT = ACT_0 \times \ldots \times ACT_n$$

sei eine Übergangsfunktion $\tau(a) : G \to G$ auf G zugeordnet; der "Null-Aktion" $(\Lambda, \ldots, \Lambda)$ die Identität (**no-op**)

$$\tau(\Lambda, \ldots, \Lambda)(s) := s \quad \text{für alle } s \in G.$$

Beispiel: In Nachrichtenaustausch-Systemen sei
$ACT_0 = \{ deliver_i(\mu, j), go_i, nogo_i \mid i, j = 1, \ldots, n, \mu \in MSG \}$ und
$ACT_i = INT_i \cup \{ send(\mu, j) \mid j = 1, \ldots, n, \mu \in MSG \}$ für $0 < i$.

Protokolle

Ein Protokoll legt die in einem Zustand möglichen Aktionen von Agenten fest.

Definition

Sei ACT_i die Menge alle für Agent i möglichen Aktionen. Ein **Protokoll** für i ist eine Funktion $P_i : L_i \to \mathcal{P}(ACT_i) \setminus \emptyset$.

P_i ist **deterministisch**, wenn $|P_i(s_i)| = 1$ für alle $s_i \in L_i$.

Ein **gemeinsames Protokoll** $P = (P_1, \ldots, P_n)$ besteht aus je einem Protokoll für jeden Agenten ($\neq 0$).

Kontexte

Das Verhalten eines Systems hängt nicht nur vom möglichen Verhalten der Agenten ab, was durch ein gemeinsames Protokoll beschrieben ist, sondern i.a. auch von

- dem Verhalten der Umgebung 0, bzw. einem Protokoll P_0,
- einer Transition $\tau : ACT \rightarrow (G \rightarrow G)$, die die Auswirkung gemeinsamer Aktionen festlegt,
- den möglichen Anfangszuständen $G_0 \subseteq G$,
- globalen Bedingungen an "akzeptable" Abläufe, $\Psi \subseteq \mathcal{R}$.

Globale Bedingungen lassen sich nicht immer durch Protokolle formulieren, z.B. Fairnessbedingungen wie

$$Fair := \Box\Diamond(send(\mu, j, i)) \rightarrow \Diamond(recieve(\mu, i, j)),$$

nach denen auf jedem Lauf ein Ereignis irgendwann eintritt.

Definition

Ein **Kontext** $\gamma = (P_0, G_0, \tau, \Psi)$ zu G und ACT besteht aus

- einem Protokoll $P_0 : L_0 \to \mathcal{P}(ACT_0) \setminus \{\emptyset\}$ der Umgebung,
- einer Menge $G_0 \subseteq G$ von globalen Anfangszuständen,
- einer Transition $\tau : ACT \to (G \to G)$,
- einer Zulässigkeitsbedingung $\Psi \subseteq \mathbb{N} \to G$ an Abläufe.

Beispiel: ein **speichernder Kontext** ist ein Kontext, wo

- die Umgebungszustände die erfolgten gemeinsamen Aktionen speichern, $L_0 \subseteq ACT^*$,
- in Anfangszuständen die Umgebung noch keine gemeinsame Aktion speichert: für $s \in G_0$ ist $s_0 = \langle\rangle$,
- bei $\tau(a)(s) = s'$ stets $s'_0 = s_0 \cdot a$ ist.

- G_0 beschreibt die Anfangsbedingungen,
- P_0 und τ beschreiben das lokale Verhalten,
- Ψ beschränkt das globale Verhalten.

Definition

Ein **interpretierter Kontext** (γ, π) ist eine Kontext γ mit einer Interpretation $\pi : G \times \Phi \longrightarrow \{\textbf{true}, \textbf{false}\}$.

Definition

Ein Lauf $r : \mathbb{N} \to G$ ist **verträglich mit dem Protokoll** $P = (P_1, \ldots, P_n)$ **im Kontext** $\gamma = (P_0, G_0, \tau, \Psi)$, wenn

1. der Lauf r beginnt in einem Anfangszustand: $r(0) \in G_0$,
2. jeder Nachfolgezustand in r entsteht durch eine von P und γ erlaubte gemeinsame Aktion: zu $r(m) = (s_0, \ldots, s_n)$ ist

$$r(m+1) = \tau(a)(r(m))$$

   für ein $a \in P_0(s_0) \times \ldots \times P_n(s_n)$.
3. r ist zulässig, d.h. $r \in \Psi$.

Sind nur Bedingungen 1. und 2. erfüllt, heißt r mit P und γ **schwach verträglich**.
Gibt es ein r mit 1.-3., so heißt P mit γ **verträglich**.

Definition

Das System $\mathcal{R}$ **repräsentiert** P **im Kontext** γ, wenn

$$\mathcal{R} = \mathcal{R}^{rep}(P, \gamma) := \{\, r : \mathbb{N} \to G \mid r \text{ ist mit } P \text{ in } \gamma \text{ verträglich} \,\}$$

Ein System $\mathcal{R}$ ist **mit** P **in** γ **verträglich**, wenn $\mathcal{R} \subseteq \mathcal{R}^{rep}(P, \gamma)$.
Analog:

$$\mathcal{I}^{rep}(P, \gamma, \pi) := (\mathcal{R}^{rep}(P, \gamma), \pi).$$

Example

Repräsentation asynchroner Nachrichtensysteme $\mathcal{R}(V_1 \ldots V_n)$.
Kontext: $\gamma^{amp} = (P_0^{amp}, G_0, \tau, \textbf{true})$ mit $G_0 = \{\langle\rangle\} \times \Sigma_1 \times \ldots \times \Sigma_n$ und $deliver_i(\mu, j) \in P_0^{amp}(s_0)$ nur, wenn μ vorher von i an j geschickt, aber noch nicht ausgeliefert wurde. Für $i > 0$ sei
$P_i(h) := \{\, a \in ACT_i \mid h + a \in V_i \,\}$ ...

Nachrichtenauslieferungskontexte

Definition

(γ, π) ist ein **Nachrichtenauslieferungskontext**, wenn gilt:

- γ ist ein speichernder Kontext, d.h. s_0 umfaßt die Folge der getätigten gemeinsamen Aktionen,
- ACT_0 enthält Aktionen $deliver_i(\mu, j)$ für $\mu \in MSG$, $1 \leq i \neq j$,
- Φ enthält eine Aussage *delivered*, mit

$$\pi(s, delivered) = 1 \ :\Longleftrightarrow$$

$$s_0 = \langle \ldots, a, \ldots \rangle \text{ und es gibt } \mu, i, j \text{ mit } deliver_i(\mu, j) \in a_0.$$

Nachrichtenauslieferungssystem

Definition

Ein **Nachrichtenauslieferungssystem** ist ein System

$$\mathcal{I} = (\mathcal{R}, \pi) = \mathcal{I}^{rep}(P, \gamma, \pi),$$

wo (γ, π) ein Nachrichtenauslieferungskontext ist und P ein mit γ verträgliches Protokoll ist.

Beachte: In einem Nachrichtenauslieferungssystem $\mathcal{I}$ ist

$$M_{\mathcal{I}}, r, 0 \models \neg \mathit{delivered}.$$

Ist $\mathcal{I}$ ein *asynchrones* Nachrichtenauslieferungssystem, so folgt

$$M_{\mathcal{I}}, r, m \models \neg C(\mathit{delivered}).$$

Unzuverlässige synchrone Nachrichtensysteme

Behauptung: Auch in „hinreichend unzuverlässigen“ synchronen Nachrichtenauslieferungssystemen kann kein geteiltes Wissen über eine Nachrichtenauslieferung entstehen.

Definition

Sei $(\mathcal{R}, \pi)$ ein Nachrichtenauslieferungssystem.

$d(r, m) :=$ Anzahl der in r bis Runde m erfolgten Auslieferungen

$\mathcal{R}$ **zeigt keine Beschränkung der Lieferzeit** (umd), wenn für alle (r, m) mit $d(r, m) > 0$ es Agenten i und Läufe $r' \in \mathcal{R}$ gibt mit

1. für alle Agenten $j \neq i$ und Runden $m' \leq m$ ist $r'_j(m') = r_j(m')$,
2. $d(r', m) < d(r, m)$,

d.h. nur Agent i weiß, daß er in r bis m die letzte Nachricht erhielt.

Theorem

Sei $\mathcal{I} = (\mathcal{R}, \pi)$ ein Nachrichtenauslieferungssystem, wo $\mathcal{R}$ keine Beschränkung der Lieferzeit hat, aber ≥ 2 Agenten. Dann ist

$$\mathcal{I} \models \neg C(\mathit{delivered}).$$

Beweis: Sei (r, m) ein Punkt mit $d(r, m) > 0$, und für alle (r', m') mit $d(r', m') < d(r, m)$ sei schon $\mathcal{I}, r', m' \models \neg C(\mathit{delivered})$.

Wähle i und r' wie in der umd-Bedingung, also $d(r', m) < d(r, m)$. Nach Induktion ist $\mathcal{I}, r', m \models \neg C(\mathit{delivered})$. Für $j \neq i$ ist $(r', m) \sim_j (r, m)$, also

$$\mathcal{I}, r, m \models \neg C(\mathit{delivered}),$$

da für alle φ

$$\mathcal{I} \models C(\varphi) \rightarrow K_j(C(\varphi)).$$

Koordinierung von Handlungen

Definition

Ein **ca-verträglicher Kontext** ist ein Nachrichtenauslieferungskontext (γ, π), sodaß für $i = 1, 2$:

- $attack_i \in ACT_i$,
- $nogo_i \notin ACT_0$,
- $attacked_i \in \Phi$ und für alle $s \in G$ ist

$$\pi(s, attacked_i) = 1 \ :\Longleftrightarrow \text{ es gibt } a, b \in ACT \text{ mit}$$
$$s_0 = \langle \ldots, a, b, \ldots \rangle \text{ mit } a_i \ni attack_i \text{ und } a_0 \ni go_i.$$
$$\text{und „} s_i = \langle \ldots, a_i, b_i, \ldots \rangle \text{“ (sodaß } i \text{ weiß, was er getan hat)}$$

Abkürzungen:

$$attacking_i \ :\Longleftrightarrow \ \neg attacked_i \wedge \bigcirc attacked_i$$
$$attacking \ :\Longleftrightarrow \ attacking_1 \wedge attacking_2$$

Definition

Die Spezifikation σ^{ca} besteht aus allen ca-verträglichen Systemen $\mathcal{I}$ mit

1. $\mathcal{I} \models \mathit{attacking}_1 \leftrightarrow \mathit{attacking}_2$,
2. $\mathcal{I} \models \neg \mathit{delivered} \rightarrow \neg \mathit{attacking}$,
3. $\mathcal{I}, r, m \models \mathit{attacking}$ für mindestens einen Punkt (r, m) von $\mathcal{I}$.

Das entspricht dem einleitenden Problem:

1. beide Generäle greifen höchstens gleichzeitig an,
2. ohne Absprache findet kein Angriff statt,
3. irgendwann soll ein Angriff stattfinden.

Theorem

Sei (γ, π) ein ca-verträglicher Kontext und P ein deterministisches Protokoll. Ist $\mathcal{I} = \mathcal{I}^{rep}(P, \gamma, \pi) \in \sigma^{ca}$, so gilt

$$\mathcal{I} \models attacking \rightarrow C(attacking).$$

Beweis: Die Behauptung folgt mit der Induktionsregel aus

$$\mathcal{I} \models attacking \rightarrow E(attacking). \qquad (1)$$

Sei $\mathcal{I}, r, m \models attacking_i$ und $(r', m') \sim_i (r, m)$.

Da $\mathcal{I}, r, m \models \neg attacked_i$ und γ ca-verträglich ist, ist
$\langle \ldots, attack_i, \ldots \rangle \neq r_i(m) = r'_i(m')$, also $\mathcal{I}, r', m' \models \neg attacked_i$.
Wegen $P_i(r'_i(m')) = P_i(r_i(m)) = attack_i$ und $nogo_i \notin ACT_0$ ist
$\mathcal{I}, r', m' \models attacking_i$, und wegen $\mathcal{I} \in \sigma^{ca}$ ist
$\mathcal{I}, r', m' \models attacking_{3-i}$, also $\mathcal{I}, r', m' \models attacking$.
Also $\mathcal{I}, r, m \models K_i(attacking)$. Da r, m, i beliebig waren, gilt (1).

- Wenn $nogo_i \in ACT_0$, ist i.a.

$$\mathcal{I}^{rep}(P, \gamma, \pi) \not\models attacking_i \rightarrow K_i(attacking_i).$$

- Wenn P nicht deterministisch ist, ist i.a.

$$\mathcal{I}^{rep}(P, \gamma, \pi) \not\models attacking_i \rightarrow K_i(attacking_i).$$

Denn $\mathcal{I}, r, m \models attacking_i$ liefert zwar $attack_i \in P(r_i(m))$, aber nicht mehr $attack_i \in P(r'_i(m'))$ für $(r', m') \sim_i (r, m)$. Und $attack_i$ ist erst in $r_i(m+1) \not\sim_i r'_i(m'+1)$ gespeichert.

Abkürzung: $attacked \;:\Longleftrightarrow\; attacked_1 \wedge attacked_2$

Lemma

Sei (γ, π) ca-verträglich, P ein Protokoll und $\mathcal{I} = \mathcal{I}^{rep}(P, \gamma, \pi) \in \sigma^{ca}$. Dann ist

$$\mathcal{I} \models attacked \rightarrow C(attacked).$$

Corollary

Für $\mathcal{I}$ wie oben: $\mathcal{I} \models attacked \rightarrow C(delivered)$.

Bew.: Nach σ^{ca} ist $\mathcal{I} \models attacking \rightarrow delivered$ und $\mathcal{I} \models attacking_1 \leftrightarrow attacking_2$.
Ist $\mathcal{I}, r, m \models attacked$, so ist für ein $m' \leq m$ also

$$\mathcal{I}, r, m' \models attacking_1 \wedge attacking_2,$$

also $\mathcal{I}, r, m' \models delivered$ und $\mathcal{I}, r, m \models delivered$. Es folgt

$$\mathcal{I} \models attacked \rightarrow delivered.$$

Mit Lemma und Notwendigkeitsregel folgt die Behauptung.