Globale Redundanz: Simplifikations- und Löschregeln

Globale Redundanz:

- viele Beweisversuche lassen sich nicht zu einem Beweis fortsetzen: Sackgassen
- ein Beweisversuch kann einen anderen subsumieren

Globale Redundanz: Simplifikations- und Löschregeln

Simplifikations- und Löschregeln:

- Tautologieelimination

$$N \cup \{C \vee A \vee \neg A\} \ \triangleright \ N$$

- Subsumption

$$N \cup \{C, D\} \ \triangleright \ N \cup \{C\}$$

  falls $C\sigma \subseteq D$ (d.h. C subsumiert D),
  aber $D\tau \neq C$, für alle τ (d.h. Subsumption ist strikt).

- Reduktion (siehe unten)

Widerspruchsvollständigkeit wird bewahrt (siehe unten).

Resolutionsbeweiser RP

3 Klauselmengen: N(ew) mit neuen Resolventen
P(rocessed) enthält die simplifizierten Resolventen
O(ld) Inferenzen zwischen diesen Klauseln sind berechnet

Suchstrategie: Inferenzen werden nur gerechnet, wenn es keine Simplifikationsmöglichkeit gibt.

Satz 2.39

$$N \models \bot \;\Leftrightarrow\; N \mid \emptyset \mid \emptyset \quad \overset{*}{\triangleright} \quad N' \cup \{\bot\} \mid \_ \mid \_$$

Beweis in Bachmair, Ganzinger: Resolution Theorem Proving

Später betrachten wir die wesentliche Grundlage hierfür, den Redundanzbegriff.

Tautology elimination

$\boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \mid \boldsymbol{O} \quad \triangleright \quad \boldsymbol{N} \mid \boldsymbol{P} \mid \boldsymbol{O}$

if C is a tautology

Forward subsumption

$\boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \mid \boldsymbol{O} \quad \triangleright \quad \boldsymbol{N} \mid \boldsymbol{P} \mid \boldsymbol{O}$

if some $D \in \boldsymbol{P} \cup \boldsymbol{O}$ subsumes C

Backward subsumption

$\boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \cup \{D\} \mid \boldsymbol{O} \quad \triangleright \quad \boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \mid \boldsymbol{O}$

$\boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \mid \boldsymbol{O} \cup \{D\} \quad \triangleright \quad \boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \mid \boldsymbol{O}$

if C strictly subsumes D

Forward reduction

$\boldsymbol{N} \cup \{C \vee L\} \mid \boldsymbol{P} \mid \boldsymbol{O} \quad \triangleright \quad \boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \mid \boldsymbol{O}$

if there exists $D \vee L' \in \boldsymbol{P} \cup \boldsymbol{O}$ such that $\overline{L} = L'\sigma$ and $D\sigma \subseteq C$

Backward reduction

$\boldsymbol{N} \mid \boldsymbol{P} \cup \{C \vee L\} \mid \boldsymbol{O} \quad \triangleright \quad \boldsymbol{N} \mid \boldsymbol{P} \cup \{C\} \mid \boldsymbol{O}$

$\boldsymbol{N} \mid \boldsymbol{P} \mid \boldsymbol{O} \cup \{C \vee L\} \quad \triangleright \quad \boldsymbol{N} \mid \boldsymbol{P} \mid \boldsymbol{O} \cup \{C\}$

if there exists $D \vee L' \in \boldsymbol{N}$ such that $\overline{L} = L'\sigma$ and $D\sigma \subseteq C$

Clause processing

$\boldsymbol{N} \cup \{C\} \mid \boldsymbol{P} \mid \boldsymbol{O} \quad \triangleright \quad \boldsymbol{N} \mid \boldsymbol{P} \cup \{C\} \mid \boldsymbol{O}$

Inference computation

$\emptyset \mid \boldsymbol{P} \cup \{C\} \mid \boldsymbol{O} \quad \triangleright \quad \boldsymbol{N} \mid \boldsymbol{P} \mid \boldsymbol{O} \cup \{C\}$, mit $\boldsymbol{N} = Res_S^{\succ}(\boldsymbol{O} \cup \{C\})$

Formaler Redundanzbegriff

Sei N Menge von Grundklauseln und C eine Grundklausel (nicht notwendig in N).

$$C \text{ heißt redundant in } N \;:\Leftrightarrow\; \exists C_1, \ldots, C_n \in N : C_i \prec C \text{ und } C_1, \ldots, C_n \models C$$

Redundanz für allgemeine Klauseln:

$$C \text{ heißt redundant in } N \;:\Leftrightarrow\; C\sigma \text{ redundant in } G_\Sigma(N), \text{ für alle Grundinstanzen } C\sigma \text{ von } C$$

Intuition: Redundante Klauseln sind keine minimalen Gegenbeispiele für keine Interpretation

NB: derselbe „Ordnungsparameter" $\succ$ für Ordnungseinschränkungen und Redundanzbegriff.

Wichtige Anwendungsbeispiele

Proposition 2.40 • *C Tautologie (d.h. $\models C$) $\Rightarrow$ C redundant in jeder Menge N.*

- *$C\sigma \subset D$ $\Rightarrow$ D redundant in $N \cup \{C\}$ (strikte Subsumption: $N \cup \{C, D\} \rhd N \cup \{C\}$)*

- *$C\sigma \subseteq D$ $\Rightarrow$ $D \vee \overline{L}\sigma$ redundant in $N \cup \{C \vee L, D\}$ (Subsumptionsresolution: $N \cup \{C \vee L, D \vee \overline{L}\sigma\} \rhd N \cup \{C \vee L, D\}$)*

In vielen Fällen kann "$\subset$" zu "$\subseteq$" abgeschwächt werden.

Saturation bis auf Redundanz

N heißt saturiert bis auf Redundanz (bzgl. $Res_S^{\succ}$)

$$:\Leftrightarrow \; Res_S^{\succ}(N \setminus Red(N)) \subseteq N \cup Red(N)$$

Satz 2.41 *Sei N saturiert bis auf Redundanz. Dann:*

$$N \models \bot \;\Leftrightarrow\; \bot \in N$$

Beweis: [Skizze]

(i) Grundklauseln: betrachte die Modellkonstruktion $I_N^{\succ}$ für $Res_S^{\succ}$.
Redundante Klauseln in N sind: (a) nie produktiv; (b) keine minimale Gegenbeispiele für $I_N^{\succ}$
Die Prämissen notwendiger Inferenzen sind entweder minimale Gegenbeispiele oder produktiv.

(ii) Lifting: keine zusätzlichen Probleme im Vergleich mit dem Beweis von Satz 2.38.

Monotonieeigenschaften von Redundanz

Satz 2.42 *(i)* $N \subseteq M \;\Rightarrow\; \mathit{Red}(N) \subseteq \mathit{Red}(M)$

(ii) $M \subseteq \mathit{Red}(N) \;\Rightarrow\; \mathit{Red}(N) \subseteq \mathit{Red}(N \setminus M)$

Beweis einfach.

Damit: Redundanz bleibt bewahrt, wenn man während eines Beweisprozesses neue Klauseln hinzuableitet oder redundante Klauseln löscht.

Die Sätze 2.41 und 2.42 ist die wesentliche Grundlage für die Vollständigkeit des Beweisers RP.

Hyperresolution (Robinson 65)

Hier definieren wir eine verbesserte Variante mit Ordnungseinschränkungen und Selektion. Wie bei *Res* ist der Kalkül durch eine Atomordnung $\succ$ und eine Selektionsfunktion S parametrisiert.

$$\frac{C_1 \vee A_1 \quad \ldots \quad C_n \vee A_n \qquad \neg B_1 \vee \ldots \vee \neg B_n \vee D}{(C_1 \vee \ldots \vee C_n \vee D)\sigma}$$

mit $\sigma = \mathrm{mgu}(A_1 \doteq B_1, \ldots, A_n \doteq B_n)$, falls

(i) $A_i\sigma$ strikt maximal bzgl. $C_i\sigma$, $1 \leq i \leq n$;

(ii) nichts selektiert in C_i;

(iii) die Auftreten der $\neg B_i$ sind gerade die bzgl. S selektierten oder nichts ist selektiert in der letzten Prämisse und $n = 1$ und $\neg B_1\sigma$ maximal bzgl. $D\sigma$.

+ Faktorisieren wie bei $Res_S^{\succ}$

Hyperresolution (Robinson 65)

Hyperresolution kann durch iterierte binäre Resolution simuliert werden. Allerdings entstehen dabei Zwischenergebnisse, die bei HR nicht entstehen, die insbesondere nicht alle zu HR-Inferenzen erweitert werden können.

Es gibt viele weitere Varianten von Resolution,
vgl. Bachmair, Ganzinger: Resolution Theorem Proving.

Craig-Interpolation

Eine einfache theoretische Anwendung der Vollständigkeit von geordneter Resolution ist Craig-Interpolation:

Satz 2.43 (Craig 57) *Seien F und G propositionale Formeln so daß $F \models G$. Dann gibt es eine Formel H (genannt der Interpolant für $F \models G$), so daß H nur prop. Variablen enthält, die sowohl in F als auch in G vorkommen, und daß $F \models H$ und $H \models G$ gilt.*

Craig-Interpolation

Beweis. Bringe F und $\neg G$ in KNF. Die entstehenden Klauselmengen seien N bzw. M. Wähle eine Atomordnung $\succ$, in der Aussagenvariablen, die in F aber nicht in G vorkommen, maximal sind. Saturiere N zu N^* unter $Res_S^{\succ}$ mit leerer Selektionsfunktion S . Wenn man anschließend $N^* \cup M$ unter $Res_S^{\succ}$ saturiert, um $\bot$ abzuleiten, sind, weil N^* bereits saturiert, und wegen der Ordnungseinschränkungen nur noch solche Inferenzen irredundant, in denen etwaige Prämissen aus N^* nur Symbole enthalten, die in G vorkommen. Die Konjunktion dieser Prämissen ist ein Interpolant H. $\Box$

Das Theorem gilt entsprechend auch für allgemeine Formeln erster Stufe, ist hier aber mit Resolutionstechnologie wegen der benötigten Skolemisierung nicht so einfach zu beweisen.

1.10 Anwendungsbeispiel: Neuman-Stubblebine-Schlüsselaustauschprotokoll

- Formalisierung eines Anwendungsbeispiels
- Stand der Kunst beim automatischen Beweisen
- Beweis durch Konsistenznachweis:
  konsistent $\Rightarrow$ Unbeweisbarkeit einer Fehlermöglichkeit
- Termination braucht gute Redundanzelimination

Neuman-Stubblebine-Schlüsselaustauschprotokoll

Automatic Analysis of Security Protocols using SPASS: An Automated Theorem Prover for First-Order Logic with Equality

by Christoph Weidenbach

Sicherheitsprotokolle

Ziel: zwei Personen (Alice und Bob) wollen miteinander kommunizieren

- über ein **unsicheres** Daten- oder Telefonnetz,
- **sicher**, d. h., ohne daß ein Eindringling (Charlie) mithören oder sich als Alice oder Bob ausgeben kann.

Hilfsmittel: Verschlüsselung

- Alice und Bob vereinbaren einen gemeinsamen Schlüssel und nutzen ihn, um ihr Gespräch zu verschlüsseln.
- Nur wer den Schlüssel kennt, kann das Gespräch entschlüsseln.

Sicherheitsprotokolle

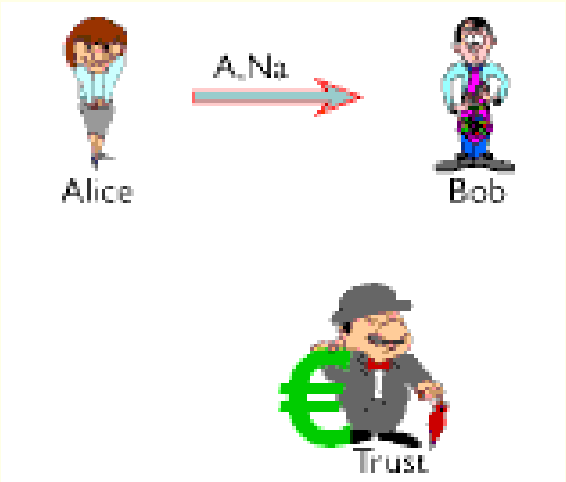
Problem: wie kommen die Gesprächspartner an den gemeinsamen Schlüssel?

- Persönliche Übergabe kommt nicht immer in Frage.
- Wird der gemeinsame Schlüssel über das Netz unverschlüsselt verschickt, könnte Charlie ihn abfangen oder austauschen.
- Annahme: es gibt eine sichere Schlüsselzentrale, mit der Alice und Bob jeweils einen gemeinsamen Schlüssel vereinbart haben.

Sicherheitsprotokolle

Das folgende Schlüsselaustauschverfahren wurde 1993 von den beiden Kryptographen Neuman und Stubblebine vorgeschlagen:

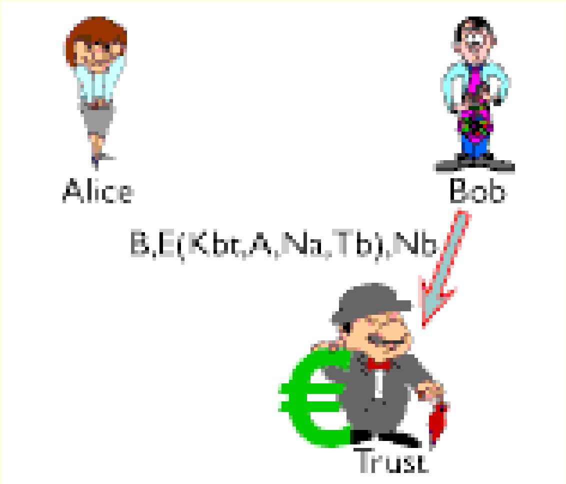
Schritt 1:



Alice schickt (offen) Identifikation und Zufallszahl an Bob.

Sicherheitsprotokolle

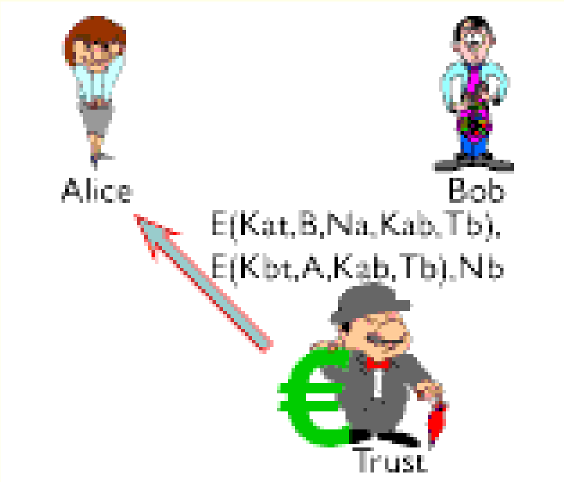
Schritt 2:



Bob leitet Nachricht weiter an Schlüsselzentrale („Trust“).

Sicherheitsprotokolle

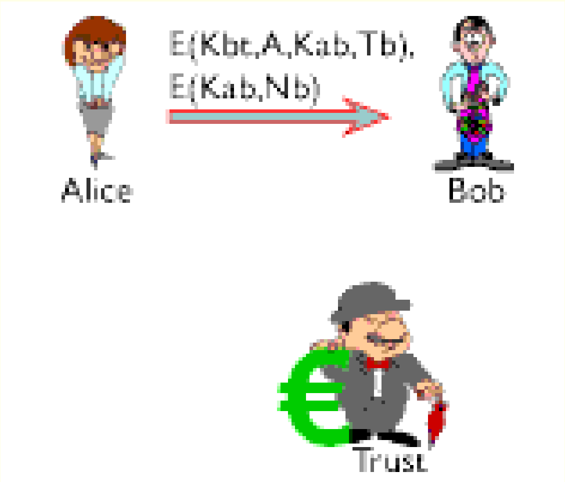
Schritt 3:



Trust schickt Nachricht an Alice. Darin: ein neuer gemeinsamer Schlüssel, einmal für Alice und einmal für Bob verschlüsselt.

Sicherheitsprotokolle

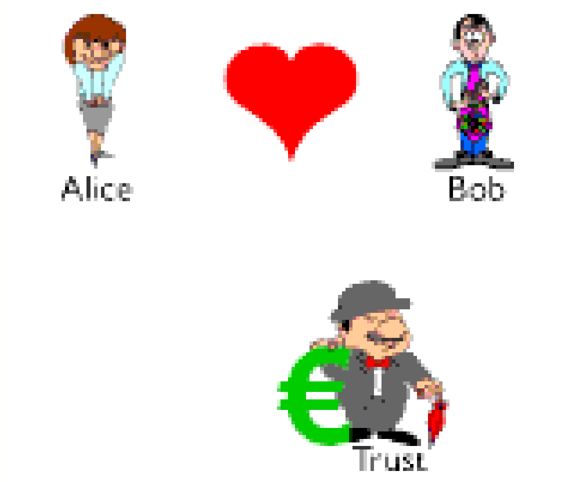
Schritt 4:



Alice leitet den neuen gemeinsamen Schlüssel weiter an Bob.

Sicherheitsprotokolle

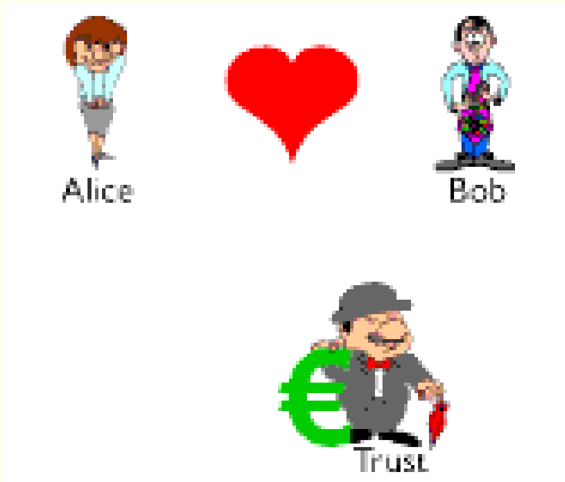
Schritt 5:



Alice und Bob können nun mit dem gemeinsamen Schlüssel kommunizieren.

Sicherheitsprotokolle

Ist das Verfahren sicher?



Wir übersetzen das Problem wieder in Formeln und lassen sie von einem Theorembeweiser untersuchen.

Sicherheitsprotokolle

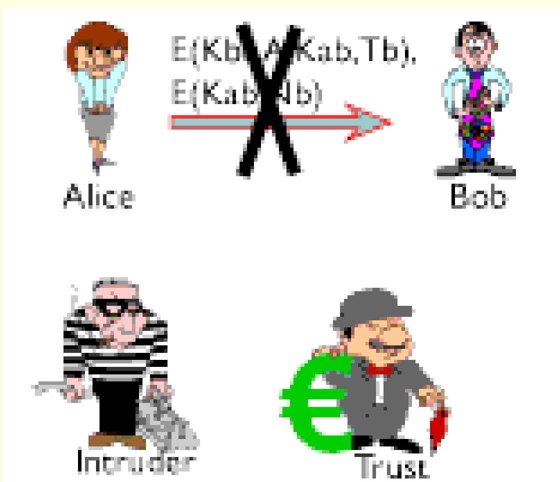
Zuerst formalisieren wir die Eigenschaften des Protokolls:

- Wenn Alice/Bob/Trust eine Nachricht in einem bestimmten Format bekommt, dann schickt er/sie eine andere Nachricht ab.
- Wenn eine Nachricht übermittelt wird, kann Charlie sie mithören.
- Wenn Charlie eine verschlüsselte Nachricht bekommt und den passenden Schlüssel hat, kann er sie entschlüsseln.
- Wenn Charlie eine Nachricht hat, dann kann er sie an Alice/Bob/Trust abschicken.

...

Sicherheitsprotokolle

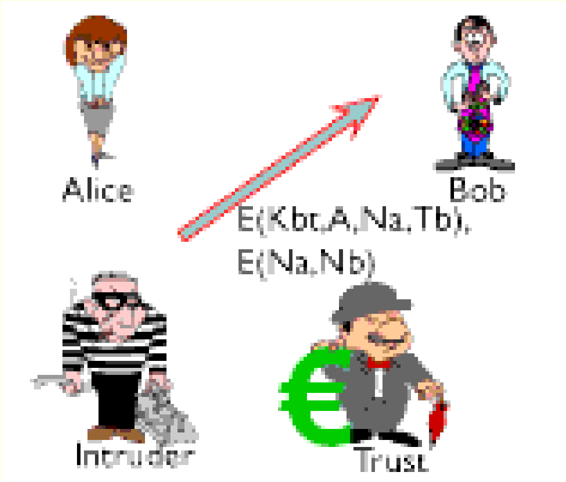
Was kann passieren?



Charlie fängt die letzte Nachricht von Alice ab.

Sicherheitsprotokolle

Was kann passieren?



Charlie schickt eine veränderte Nachricht an Bob.

Sicherheitsprotokolle

Was kann passieren?

Bob entschlüsselt die Nachricht und denkt, sie komme von Alice.

Sicherheitsprotokolle

Was kann passieren?

Bob startet Kommunikation mit Alice . . .

Sicherheitsprotokolle

Was kann passieren?

... aber spricht in Wirklichkeit mit Charlie.

Zusammenfassung: Resolutionsbeweisen

- Resolution ist ein reines Maschinenverfahren.
- geschickte Verschränkung der Aufzählung von Grundinstanzen und Nachweis von Unerfüllbarkeit durch Verwendung von Unifikation
- Parameter Atomordnung $\succ$ und Selektionsfunktion S; approximiertes Lösen der Ordnungseinschränkung auf Nichtgrundebene
- Vollständigkeitsbeweis durch Konstruktion von Modellkandidaten aus reduktiven Klauseln $C \vee A$, $A \succ C$; Inferenzen mit diesen reduzieren Gegenbeispiele.
- Einschränkung der Inferenzen lokal durch $\succ$ und S

$\Rightarrow$ weniger Beweisvarianten

- Globale Beschränkungen durch Redundanzelimination
  $\Rightarrow$ Rechnen mit “kleinen” inkonsistenten Teilmengen;
  $\Rightarrow$ Termination auf vielen entscheidbaren Fragmenten
- Trotz allem schlecht bei Ordnungen, Gleichheit und spezielleren algebraischen Theorien (Verbände, abelsche Gruppen, Ringe, Körper)
  $\Rightarrow$ weitere Spezialisierung von Inferenzsystemen nötig.