Inferring Dynamic Credentials for Rôle-based Trust Management

Vladimiro Sassone

ECS, University of Southampton

joint work with D. Gorla (Roma) and M. Hennessy (Sussex)

PPDP’06, Venice 11 July 2006
1. Rôle-based trust-management
2. RT₀ operational semantics
3. Context-dependent credentials (CDCs)
4. An enhanced inference system for CDCs
5. Inferring time validity and environmental credentials
6. Conclusions
Trust Management

- **Trust-management**: a form of distributed access control based on policy statements made by multiple principals.

- A key aspect is **delegation**: transfer of limited authority on some resources to other principals.

  Usually, this is done by means of **credentials**.

- Decisions are made according to the identity of the resource requester.

  **PROBLEM**: when resource owner and requester are unknown to each other, such a form of access control does not work.

  Must shift the focus on the **certificates** it demonstrably holds.
Trust Management

- **Trust-management**: a form of distributed access control based on policy statements made by multiple principals.

- A key aspect is **delegation**: transfer of limited authority on some resources to other principals. Usually, this is done by means of **credentials**.

- Decisions are made according to the identity of the resource requester.

  **PROBLEM**: when resource owner and requester are unknown to each other, such a form of access control does not work. Must shift the focus on the **certificates** it demonstrably holds.
Rôle-based Trust-management

AN APPROACH: RT (Li, Mitchell, Winsborough@IEEE-SSP02)

- Trust management + rôle-based access control
- Inspired by trust-management languages such as SPKI/SDSI
- Includes basic operations to perform complex forms of delegation
- A family of increasingly powerful languages, $RT_0$ being the basic form.
An auditor can inspect an enterprise **Ent** only if is authorised by the UK government: `Ent.auditor ← UK.auditor`;

An auditor is authorised if is a member of a government recognised society: `UK.auditor ← UK.authSoc.member`;

Auditing societies must be legally registered and ‘fair’:

```
```

Assume **BSoc** is both legally registered and ‘fair’ for UK law:

```
UK.legalSoc ← BSoc and UK.fairSoc ← BSoc;
```

and that **B** belongs to **BSoc**: `BSoc.member ← B`;

From this, we want to infer that **B** can inspect **Ent**.
RT₀, by example

- An auditor can inspect an enterprise \texttt{ENT} only if is authorised by the UK government: \texttt{ENT.auditor ← UK.auditor};

- An auditor is authorised if is a member of a government recognised society: \texttt{UK.auditor ← UK.authSoc.member};

- Auditing societies must be legally registered and ‘fair’: \texttt{UK.authSoc ← UK.legalSoc ⊓ UK.fairSoc}.

- Assume \texttt{BSoc} is both legally registered and ‘fair’ for UK law:
  \texttt{UK.legalSoc ← BSoc} and \texttt{UK.fairSoc ← BSoc};

  and that \texttt{B} belongs to \texttt{BSoc}: \texttt{BSoc.member ← B};

- From this, we want to infer that \texttt{B} can inspect \texttt{ENT}. 

RT₀, by example

- An auditor can inspect an enterprise \textbf{Ent} only if is authorised by the UK government: \textit{Ent.auditor} \leftarrow \textit{UK.auditor};

- An auditor is authorised if is a member of a government recognised society: \textit{UK.auditor} \leftarrow \textit{UK.authSoc.member};

- Auditing societies must be legally registered and ‘fair’: \textit{UK.authSoc} \leftarrow \textit{UK.legalSoc} \sqcap \textit{UK.fairSoc}.

- Assume \textbf{BSoc} is both legally registered and ‘fair’ for UK law:

  \textit{UK.legalSoc} \leftarrow \textbf{BSoc} \quad \text{and} \quad \textit{UK.fairSoc} \leftarrow \textbf{BSoc};

  and that \textbf{B} belongs to \textbf{BSoc}: \textbf{BSoc.member} \leftarrow \textbf{B};

- From this, we want to infer that \textbf{B} can inspect \textbf{Ent}. 
RT₀, by example

- An auditor can inspect an enterprise \( E \) only if is authorised by the UK government: \( E . \text{auditor} \leftarrow UK.\text{auditor} \);

- An auditor is authorised if is a member of a government recognised society: \( UK.\text{auditor} \leftarrow UK.\text{authSoc}.\text{member} \);

- Auditing societies must be legally registered and ‘fair’: \( UK.\text{authSoc} \leftarrow UK.\text{legalSoc} \sqcap UK.\text{fairSoc} \).

- Assume \( BSoc \) is both legally registered and ‘fair’ for UK law: \( UK.\text{legalSoc} \leftarrow BSoc \) and \( UK.\text{fairSoc} \leftarrow BSoc \);

- and that \( B \) belongs to \( BSoc \): \( BSoc.\text{member} \leftarrow B \);

- From this, we want to infer that \( B \) can inspect \( E \).
RT₀, more formally

Four kinds of RT₀-credential:

1. \( A.r \leftarrow B \) states that principal \( B \) belongs to the rôle \( r \) governed by principal \( A \);

2. \( A.r \leftarrow B.s \) states that all members of rôle \( s \) governed by \( B \) also belong to rôle \( r \) governed by \( A \);

3. \( A.r \leftarrow B.s \sqcap C.t \) states that rôle \( r \) governed by \( A \) contains all the members of both \( B \)'s rôle \( s \) and of \( C \)'s rôle \( t \);

4. \( A.r \leftarrow B.s.t \) states that rôle \( r \) governed by \( A \) contains all the members of \( C \)'s rôle \( t \), for every \( C \) belonging to \( B \)'s rôle \( s \).
**RT\(_0\), more formally**

**Four kinds of RT\(_0\)-credential:**

1. \(A.r \leftarrow B\) states that principal \(B\) belongs to the rôle \(r\) governed by principal \(A\);

2. \(A.r \leftarrow B.s\) states that all members of rôle \(s\) governed by \(B\) also belong to rôle \(r\) governed by \(A\);

3. \(A.r \leftarrow B.s \sqcap C.t\) states that rôle \(r\) governed by \(A\) contains all the members of both \(B\)'s rôle \(s\) and of \(C\)'s rôle \(t\);

4. \(A.r \leftarrow B.s.t\) states that rôle \(r\) governed by \(A\) contains all the members of \(C\)'s rôle \(t\), for every \(C\) belonging to \(B\)'s rôle \(s\).
RT₀, more formally

Four kinds of RT₀-credential:

1. A.r ← B states that principal B belongs to the rôle r governed by principal A;

2. A.r ← B.s states that all members of rôle s governed by B also belong to rôle r governed by A;

3. A.r ← B.s ∩ C.t states that rôle r governed by A contains all the members of both B’s rôle s and of C’s rôle t;

4. A.r ← B.s.t states that rôle r governed by A contains all the members of C’s rôle t, for every C belonging to B’s rôle s.
RT\(_0\), more formally

Four kinds of RT\(_0\)-credential:

1. \(A.r \leftarrow B\) states that principal \(B\) belongs to the rôle \(r\) governed by principal \(A\);

2. \(A.r \leftarrow B.s\) states that all members of rôle \(s\) governed by \(B\) also belong to rôle \(r\) governed by \(A\);

3. \(A.r \leftarrow B.s \sqcap C.t\) states that rôle \(r\) governed by \(A\) contains all the members of both \(B\)'s rôle \(s\) and of \(C\)'s rôle \(t\);

4. \(A.r \leftarrow B.s.t\) states that rôle \(r\) governed by \(A\) contains all the members of \(C\)'s rôle \(t\), for every \(C\) belonging to \(B\)'s rôle \(s\).
RT₀, more formally

**Four kinds of RT₀-credential:**

1. \( A.r \leftarrow B \) states that principal \( B \) belongs to the rôle \( r \) governed by principal \( A \);

2. \( A.r \leftarrow B.s \) states that all members of rôle \( s \) governed by \( B \) also belong to rôle \( r \) governed by \( A \);

3. \( A.r \leftarrow B.s \cap C.t \) states that rôle \( r \) governed by \( A \) contains all the members of both \( B \)'s rôle \( s \) and of \( C \)'s rôle \( t \);

4. \( A.r \leftarrow B.s.t \) states that rôle \( r \) governed by \( A \) contains all the members of \( C \)'s rôle \( t \), for every \( C \) belonging to \( B \)'s rôle \( s \).
**RT\(_0\), more formally**

Four kinds of \textbf{RT\(_0\)-credential:}\(\)

1. \(A.r \leftarrow B\) states that principal \(B\) belongs to the rôle \(r\) governed by principal \(A\);

2. \(A.r \leftarrow B.s\) states that all members of rôle \(s\) governed by \(B\) also belong to rôle \(r\) governed by \(A\);

3. \(A.r \leftarrow B.s \sqcap C.t\) states that rôle \(r\) governed by \(A\) contains all the members of both \(B\)'s rôle \(s\) and of \(C\)'s rôle \(t\);

4. \(A.r \leftarrow B.s.t\) states that rôle \(r\) governed by \(A\) contains all the members of \(C\)'s rôle \(t\), for every \(C\) belonging to \(B\)'s rôle \(s\).
An inference system for $\text{RT}_0$

$\text{RT}_0$ semantics: fixpoint construction; equivalently, translation into logic programs and minimal Herbrand models.

A more ‘operational’ flavour: certificate inference from a (finite) set of credentials $P$.

\[
\begin{align*}
c \in P & \quad P \triangleright A.r \leftarrow B.s & \quad P \triangleright B.s \leftarrow C \\
P \triangleright c & \quad P \triangleright A.r \leftarrow C \\
P \triangleright A.r \leftarrow B.s.t & \quad P \triangleright B.s \leftarrow C & \quad P \triangleright C.t \leftarrow D \\
P \triangleright A.r \leftarrow D \\
P \triangleright A.r \leftarrow B.s \cap C.t & \quad P \triangleright B.s \leftarrow D & \quad P \triangleright C.t \leftarrow D \\
P \triangleright A.r \leftarrow D
\end{align*}
\]
An inference system for $RT_0$

$RT_0$ semantics: fixpoint construction; equivalently, translation into logic programs and minimal Herbrand models.

A more ‘operational’ flavour: certificate inference from a (finite) set of credentials $P$.

\[
\begin{align*}
\text{c} & \in P \\
\hline
P \models \text{c} & \quad P \models \text{A.r} \leftarrow \text{B.s} & P \models \text{B.s} \leftarrow \text{C} \\
& \quad P \models \text{A.r} \leftarrow \text{C} \\
& \quad P \models \text{A.r} \leftarrow \text{B.s.t} & P \models \text{B.s} \leftarrow \text{C} & P \models \text{C.t} \leftarrow \text{D} \\
& \quad P \models \text{A.r} \leftarrow \text{D} \\
& \quad P \models \text{A.r} \leftarrow \text{B.s} \sqcap \text{C.t} & P \models \text{B.s} \leftarrow \text{D} & P \models \text{C.t} \leftarrow \text{D} \\
& \quad P \models \text{A.r} \leftarrow \text{D}
\end{align*}
\]
An inference system for $\text{RT}_0$

$\text{RT}_0$ semantics: fixpoint construction; equivalently, translation into logic programs and minimal Herbrand models.

A more ‘operational’ flavour: certificate inference from a (finite) set of credentials $P$.

$$
\begin{align*}
  c & \in P \\
  \frac{}{P \triangleright c} \\
  P \triangleright A.r & \leftarrow B.s \\
  P \triangleright B.s & \leftarrow C \\
  \frac{}{P \triangleright A.r \leftarrow C} \\
  & \\
  P \triangleright A.r & \leftarrow B.s \cdot t \\
  P \triangleright B.s & \leftarrow C \\
  P \triangleright C.t & \leftarrow D \\
  \frac{}{P \triangleright A.r \leftarrow D} \\
  & \\
  P \triangleright A.r & \leftarrow B.s \sqcap C.t \\
  P \triangleright B.s & \leftarrow D \\
  P \triangleright C.t & \leftarrow D \\
  \frac{}{P \triangleright A.r \leftarrow D}
\end{align*}
$$
An inference system for $\text{RT}_0$

$\text{RT}_0$ semantics: fixpoint construction; equivalently, translation into logic programs and minimal Herbrand models.

A more ‘operational’ flavour: certificate inference from a (finite) set of credentials $P$.

\[
\begin{align*}
&c \in P \\
&\quad \Rightarrow P \Rightarrow c
\end{align*}
\]

\[
\begin{align*}
P &\Rightarrow A.r \leftarrow B.s & P &\Rightarrow B.s \leftarrow C \\
&\quad \Rightarrow P \Rightarrow A.r \leftarrow C
\end{align*}
\]

\[
\begin{align*}
P &\Rightarrow A.r \leftarrow B.s.t & P &\Rightarrow B.s \leftarrow C & P &\Rightarrow C.t \leftarrow D \\
&\quad \Rightarrow P \Rightarrow A.r \leftarrow D
\end{align*}
\]

\[
\begin{align*}
P &\Rightarrow A.r \leftarrow B.s \sqcap C.t & P &\Rightarrow B.s \leftarrow D & P &\Rightarrow C.t \leftarrow D \\
&\quad \Rightarrow P \Rightarrow A.r \leftarrow D
\end{align*}
\]
An inference system for RT₀

RT₀ semantics: fixpoint construction; equivalently, translation into logic programs and minimal Herbrand models.

A more ‘operational’ flavour: certificate inference from a (finite) set of credentials $P$.

\[
\begin{align*}
\text{c} & \in P \\
\implies & \quad P \triangleright c
\end{align*}
\]

\[
\begin{align*}
P \triangleright A.r & \leftarrow B.s \\
P \triangleright B.s & \leftarrow C \\
\implies & \quad P \triangleright A.r & \leftarrow C
\end{align*}
\]

\[
\begin{align*}
P \triangleright A.r & \leftarrow B.s \cdot t \\
P \triangleright B.s & \leftarrow C \\
P \triangleright C.t & \leftarrow D \\
\implies & \quad P \triangleright A.r & \leftarrow D
\end{align*}
\]

\[
\begin{align*}
P \triangleright A.r & \leftarrow B.s \sqcap C.t \\
P \triangleright B.s & \leftarrow D \\
P \triangleright C.t & \leftarrow D \\
\implies & \quad P \triangleright A.r & \leftarrow D
\end{align*}
\]
An inference system for $RT_0$

$RT_0$ semantics: fixpoint construction; equivalently, translation into logic programs and minimal Herbrand models.

A more ‘operational’ flavour: certificate inference from a (finite) set of credentials $P$.

\[
\begin{align*}
c \in P & \quad \frac{P \triangleright A.r \leftarrow B.s}{P \triangleright c} \\
P \triangleright B.s & \quad \frac{P \triangleright B.s \leftarrow C}{P \triangleright A.r \leftarrow C} \\
P \triangleright A.r \leftarrow B.s.t & \quad P \triangleright B.s \leftarrow C \quad P \triangleright C.t \leftarrow D \\
& \quad \frac{P \triangleright A.r \leftarrow C}{P \triangleright A.r \leftarrow D} \\
P \triangleright A.r \leftarrow B.s \sqcap C.t & \quad P \triangleright B.s \leftarrow D \quad P \triangleright C.t \leftarrow D \\
& \quad \frac{P \triangleright A.r \leftarrow D}{P \triangleright A.r \leftarrow D}
\end{align*}
\]
The auditing example, formalised

Derive a credential for B as a UK Auditor:

\[ \begin{align*}
P &\rightarrow \text{UK.legalSoc} \leftarrow \text{BS} \\
& \quad P \rightarrow \text{UK.fairSoc} \leftarrow \text{BS} \\
& \quad P \rightarrow \text{UK.authSoc} \leftarrow \text{UK.legalSoc} \cap \text{UK.fairSoc} \\
& \quad P \rightarrow \text{UK.authSoc} \leftarrow \text{BS} \\
& \quad P \rightarrow \text{BS.member} \leftarrow B \\
& \quad P \rightarrow \text{UK.auditor} \leftarrow \text{UK.authSoc.member} \\
& \quad P \rightarrow \text{UK.auditor} \leftarrow B \\
\end{align*} \]

We can then derive a credential authorising B to inspect Ent:

\[ \begin{align*}
& \quad P \rightarrow \text{Ent.auditor} \leftarrow \text{UK.auditor} \\
& \quad P \rightarrow \text{UK.auditor} \leftarrow B \\
& \quad P \rightarrow \text{Ent.auditor} \leftarrow B
\end{align*} \]
The auditing example, formalised

Derive a credential for $B$ as a UK **AUDITOR**:

\[
P \triangleright \text{UK.legalSoc} \leftarrow \text{BS} \quad P \triangleright \text{UK.fairSoc} \leftarrow \text{BS} \\
P \triangleright \text{UK.authSoc} \leftarrow \text{UK.legalSoc} \sqcap \text{UK.fairSoc} \\
P \triangleright \text{UK.authSoc} \leftarrow \text{BS} \\
P \triangleright \text{BS.member} \leftarrow B \\
P \triangleright \text{UK.auditor} \leftarrow \text{UK.authSoc.member} \\
P \triangleright \text{UK.auditor} \leftarrow B
\]

We can then derive a credential authorising $B$ to inspect **Ent**:

\[
P \triangleright \text{Ent.auditor} \leftarrow \text{UK.auditor} \\
P \triangleright \text{UK.auditor} \leftarrow B \\
P \triangleright \text{Ent.auditor} \leftarrow B
\]
The auditing example, formalised

Derive a credential for $B$ as a UK $\textbf{AUDITOR}$:

\[
\begin{align*}
P \triangleright & \text{UK.legalSoc} \leftarrow \text{BS} & P \triangleright & \text{UK.fairSoc} \leftarrow \text{BS} \\
& P \triangleright \text{UK.authSoc} \leftarrow \text{UK.legalSoc} \sqcap \text{UK.fairSoc} & P \triangleright & \text{UK.authSoc} \leftarrow \text{BS} \\
& P \triangleright \text{BS.member} \leftarrow B & P \triangleright & \text{UK.auditor} \leftarrow \text{UK.authSoc.member} \\
\hline
& P \triangleright \text{UK.auditor} \leftarrow B
\end{align*}
\]

We can then derive a credential authorising $B$ to inspect $\text{Ent}$:

\[
\begin{align*}
& P \triangleright \text{Ent.auditor} \leftarrow \text{UK.auditor} & P \triangleright & \text{UK.auditor} \leftarrow B \\
\hline
& P \triangleright \text{Ent.auditor} \leftarrow B
\end{align*}
\]
The auditing example, formalised

Derive a credential for B as a UK auditor:

\[
P \triangleright UK.legalSoc \leftarrow BS \quad P \triangleright UK.fairSoc \leftarrow BS \quad P \triangleright UK.authSoc \leftarrow UK.legalSoc \cap UK.fairSoc \quad P \triangleright BS.member \leftarrow B \quad P \triangleright UK.auditor \leftarrow UK.authSoc.member
\]

We can then derive a credential authorising B to inspect Ent:

\[
P \triangleright Ent.auditor \leftarrow UK.auditor \quad P \triangleright UK.auditor \leftarrow B \quad P \triangleright Ent.auditor \leftarrow B
\]
Context-dependent credentials, informally

Extend RT₀ by adding boolean guards and time validity:
- permissions often hold only for specific periods of time;
- can be issued/revoked according to the context.

Example (auditing, revised)

- BSoc becomes legal only after its registration at time τ:
  \[
  \text{UK.legalSOC} \leftarrow \text{BSoc in } [τ, +∞)
  \]

- UK’s fairness certificates are valid only for a period of time \(v₁\), and B is a member of BSoc for a fixed period \(v₂\):
  \[
  \text{UK.fairSOC} \leftarrow \text{BSoc in } v₁, \quad \text{BSoc.member} \leftarrow B \text{ in } v₂
  \]

- B can inspect ENT if he is authorised and is not one of ENT’s employees:
  \[
  \text{if } B \in \text{UK.auditor} \land B \notin \text{ENT.employee} \text{ then ENT.auditor } \leftarrow B
  \]
Context-dependent credentials, informally

Extend RT_0 by adding **boolean guards** and **time validity**:  
- permissions often hold only for specific periods of time;  
- can be issued/revoked according to the context.

**Example (auditing, revised)**

- **BSoc** becomes legal only after its registration at time \( \tau \):
  \[
  \text{UK.legalSoc} \leftarrow \text{BSoc in } [\tau, +\infty) 
  \]

- UK’s fairness certificates are valid only for a period of time \( v_1 \), and **B** is a member of **BSoc** for a fixed period \( v_2 \):
  \[
  \text{UK.fairSoc} \leftarrow \text{BSoc in } v_1, \quad \text{BSoc.member} \leftarrow \text{B in } v_2 
  \]

- **B** can inspect **Ent** if he is authorised and is not one of **Ent**’s employees:
  \[
  \text{if } B \in \text{UK.auditor} \land B \notin \text{Ent.employee} \quad \text{then Ent.auditor} \leftarrow B 
  \]
Context-dependent credentials, informally

Extend RT₀ by adding boolean guards and time validity:
- permissions often hold only for specific periods of time;
- can be issued/revoked according to the context.

Example (auditing, revised)

- **BSoc** becomes legal only after its registration at time \( t₀ \):
  \[
  \text{UK.legalSOC} \leftarrow \text{BSoc in } [t₀, +\infty)
  \]

- UK’s fairness certificates are valid only for a period of time \( v₁ \), and **B** is a member of **BSoc** for a fixed period \( v₂ \):
  \[
  \text{UK.fairSOC} \leftarrow \text{BSoc in } v₁ , \quad \text{BSoc.member} \leftarrow \text{B in } v₂
  \]

- **B** can inspect **ENT** if he is authorised and is not one of **ENT**’s employees:
  \[
  \text{if } B \in \text{UK.auditor} \land B \notin \text{ENT.employee} \text{ then ENT.auditor} \leftarrow B
  \]
Context-dependent credentials, formally

**CDCs**

**Rôle Expressions:** \( e ::= B \mid B.s \mid B.s.t \mid B.s \sqcap C.t \)

**RT\(_0\) Credential:** \( c ::= A.r \leftarrow e \)

**Guards:** \( g ::= \top \mid B \in A.r \mid B \notin A.r \mid g_1 \land g_2 \)

**Time Validity:** \( \nu ::= [\tau_1, \tau_2] \mid [\tau_1, \tau_2) \mid (\tau_1, \tau_2] \mid (\tau_1, \tau_2) \\
\mid (\tau_1, +\infty) \mid (\tau, +\infty) \\
\mid (\tau, \infty) \mid \nu_1 \cup \nu_2 \mid \nu_1 \cap \nu_2 \mid \nu_1 \setminus \nu_2 \)

**CDCs:** \( \chi ::= \text{if } g \text{ then } c \text{ in } \nu \)
Context-dependent credentials, formally

**CDCs**

**Role Expressions:**  
\[ e ::= B \mid B.s \mid B.s.t \mid B.s \cap C.t \]

**RT\(_0\) Credential:**  
\[ c ::= A.r \leftarrow e \]

**Guards:**  
\[ g ::= \texttt{tt} \mid B \in A.r \mid B \notin A.r \mid g_1 \land g_2 \]

**Time Validity:**  
\[ \nu ::= [\tau_1, \tau_2] \mid (\tau_1, \tau_2) \mid (\tau_1, \tau_2] \mid (\tau_1, \tau_2) \]
\[ \mid (\tau, \tau] \mid (-\infty, \tau] \mid [\tau, +\infty) \mid (\tau, +\infty) \]
\[ \mid (-\infty, +\infty) \mid \nu_1 \cup \nu_2 \mid \nu_1 \cap \nu_2 \mid \nu_1 \setminus \nu_2 \]

**CDCs:**  
\[ \chi ::= \textbf{if } g \textbf{ then } c \textbf{ in } \nu \]
Context-dependent credentials, formally

**Rôle Expressions:** \( e ::= B \mid B.s \mid B.s.t \mid B.s \cap C.t \)

**RT\(_0\) Credential:** \( c ::= A.r \leftarrow e \)

**Guards:** \( g ::= \top \mid B \in A.r \mid B \notin A.r \mid g_1 \land g_2 \)

**Time Validity:** \( \nu ::= [\tau_1, \tau_2] \mid (\tau_1, \tau_2) \mid (\tau_1, \tau_2] \mid (\tau_1, \tau_2) \)

\[ \mid (\tau, +\infty) \mid (\tau_1, +\infty) \mid (\tau, +\infty) \]

\[ \mid (\tau_1, +\infty) \mid (\tau, +\infty) \mid (\tau, +\infty) \]

\[ \mid (\tau_1, +\infty) \mid \nu_1 \cup \nu_2 \mid \nu_1 \cap \nu_2 \mid \nu_1 \setminus \nu_2 \]

**CDCs:** \( \chi ::= \text{if } g \text{ then } c \text{ in } \nu \)
Context-dependent credentials, formally

**CDCs**

**Rôle Expressions:**  
\[ e ::= B \mid B.s \mid B.s.t \mid B.s \cap C.t \]

**RT₀ Credential:**  
\[ c ::= A.r \leftarrow e \]

**Guards:**  
\[ g ::= \top \mid B \in A.r \mid B \notin A.r \mid g₁ \land g₂ \]

**Time Validity:**  
\[ \nu ::= [τ₁, τ₂] \mid (τ₁, τ₂) \mid (τ₁, τ₂] \mid (τ₁, +\infty) \]
\[ \mid (−\infty, τ] \mid (−\infty, τ) \mid [τ, +\infty) \mid (τ, +\infty) \]
\[ \mid (−\infty, +\infty) \mid \nu₁ \cup \nu₂ \mid \nu₁ \cap \nu₂ \mid \nu₁ \setminus \nu₂ \]

**CDCs:**  
\[ \chi ::= \text{if } g \text{ then } c \text{ in } \nu \]
An inference system for CDCs

Given a (finite) set of CDCs $\mathbb{N}$, adapt the inference system to derive new certificates.

Judgements take the form

$\mathbb{N} \vdash_{\tau} c$

and mean that $c$ can be inferred, at time $\tau$, from $\mathbb{N}$.

This entails that $\mathbb{N}$ satisfies

- all the positive guards of the CDCs used in the inference;
- none of their negative guards.
The key rule is:

**Rules**

\[
\text{if } \bigwedge_{i} B_i \in A_i.r_i \land \bigwedge_{j} B'_j \notin A'_j.r'_j \text{ then } c \text{ in } \nu \in \mathbb{N} \\
\forall i \in \mathbb{N} \vdash_{\tau} A_i.r_i \leftarrow B_i \\
\forall j \in \mathbb{N} \not\vdash_{\tau} A'_j.r'_j \leftarrow B'_j \\
\mathbb{N} \vdash_{\tau} c
\]

To use a CDC

- all its positive guards must be inferrable,
- none of its negative guards must be inferrable, and
- the CDC must be valid at the inference time \(\tau\).
An inference system for CDCs

The other rules are adapted mutatis mutandis from those for $RT_0$:

**Rules**

\[
\begin{align*}
\Gamma \vdash \tau & \quad A.r \leftarrow B.s \\
\Gamma \vdash \tau & \quad B.s \leftarrow C \\
\Gamma \vdash \tau & \quad A.r \leftarrow C
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash \tau & \quad A.r \leftarrow B.s \cdot t \\
\Gamma \vdash \tau & \quad B.s \leftarrow C \\
\Gamma \vdash \tau & \quad C.t \leftarrow D \\
\Gamma \vdash \tau & \quad A.r \leftarrow D
\end{align*}
\]

\[
\begin{align*}
\Gamma \vdash \tau & \quad A.r \leftarrow B.s \sqcap C.t \\
\Gamma \vdash \tau & \quad B.s \leftarrow D \\
\Gamma \vdash \tau & \quad C.t \leftarrow D \\
\Gamma \vdash \tau & \quad A.r \leftarrow D
\end{align*}
\]
Technical results

**PROBLEM:** the inference system has **negative premises**, which has the potential to undermine its well-foundedness.

**SOLUTION:** use the **stable model construction** (from LP, adapted to inference systems \( \text{[BOL, GROOTE]} \)) to assign meaning to the inference system whenever possible;

Following the stable model construction, we also adapt to CDCs the two existing semantics (set-theoretic and logic programming-based) of \( \text{RT}_0 \) \( \text{[MITCHELL ET AL]} \).

**The three semantics coincide.**
**Technical results**

- **PROBLEM:** the inference system has negative premises, which has the potential to undermine its well-foundedness.

- **SOLUTION:** use the stable model construction (from LP, adapted to inference systems \(\text{Bol, Groote}\)) to assign meaning to the inference system whenever possible;

> Following the stable model construction, we also adapt to CDCs the two existing semantics (set-theoretic and logic programming-based) of \(\text{RT}_0\) \(\text{(Mitchell et al)}\).

- The three semantics coincide.
Technical results

- **PROBLEM:** the inference system has negative premises, which has the potential to undermine its well-foundedness.

- **SOLUTION:** use the stable model construction (from LP, adapted to inference systems (Bol, Groote)) to assign meaning to the inference system whenever possible;

- Following the stable model construction, we also adapt to CDCs the two existing semantics (set-theoretic and logic programming-based) of \( RT_0 \) (Mitchell et al).

- The three semantics coincide.
Technical results

- **PROBLEM:** the inference system has **negative premises**, which has the potential to undermine its well-foundedness.

- **SOLUTION:** use the **stable model construction** (from LP, adapted to inference systems (Bol, Groote)) to assign meaning to the inference system whenever possible;

  Following the stable model construction, we also adapt to CDCs the two existing semantics (set-theoretic and logic programming-based) of $RT_0$ (Mitchell et al).

- **The three semantics coincide.**
Deriving constraints on the context

CDCs require full knowledge of the context where the evaluation takes place, i.e.,

- the exact time of evaluation, and
- all the CDCs available (to ensure soundness in the presence of negative premises).

In large-scale distributed systems these pieces of information are hardly available (due to asynchrony and the co-existence of multiple administrative entities).

We enhance the inference system for CDCs to also derive constraints on the execution context that validate a given inference.
Deriving time validity

Characterise the instants when a given inference holds.

\[
\begin{align*}
\text{if } & \bigwedge_i B_i \in A_i.r_i \land \bigwedge_j B'_j \notin A'_j.r'_j \text{ then } c \text{ in } \forall i . \mathbb{N} \vdash_i A_i.r_i \leftarrow B_i \quad \forall j . \mathbb{N} \vdash_j A'_j.r'_j \leftarrow B'_j \\
& \mathbb{N} \vdash (\bigcap_i \forall_i \cap \bigcap_j \forall_j) C
\end{align*}
\]

\[
\begin{align*}
\mathbb{N} & \vdash_1 A.r \leftarrow B.s \quad \mathbb{N} \vdash_2 B.s \leftarrow C \\
& \mathbb{N} \vdash_1 \forall_1 \cap \forall_2 A.r \leftarrow C
\end{align*}
\]

\[
\begin{align*}
\mathbb{N} & \vdash_1 A.r \leftarrow B.s.t \quad \mathbb{N} \vdash_2 B.s \leftarrow C \quad \mathbb{N} \vdash_3 D.t \leftarrow D \\
& \mathbb{N} \vdash_1 \forall_1 \cap \forall_2 \cap \forall_3 A.r \leftarrow D
\end{align*}
\]

\[
\begin{align*}
\mathbb{N} & \vdash_1 A.r \leftarrow B.s \cap C.t \quad \mathbb{N} \vdash_2 B.s \leftarrow D \quad \mathbb{N} \vdash_3 C.t \leftarrow D \\
& \mathbb{N} \vdash_1 \forall_1 \cap \forall_2 \cap \forall_3 A.r \leftarrow D
\end{align*}
\]
Deriving time validity

Characterise the instants when a given inference holds.

\[
\text{if } \bigwedge_i B_i \in A_i.r_i \land \bigwedge_j B'_j \notin A'_j.r'_j \text{ then } c \text{ in } v \in \mathbb{N} \\
\forall i . \mathbb{N} \models_{v_i} A_i.r_i \leftarrow B_i \\
\forall j . \mathbb{N} \models_{v_j} A'_j.r'_j \leftarrow B'_j \\
\mathbb{N} \models_{(v \cap \cap_i v_i) \setminus \cup_j v_j} C
\]

\[
\mathbb{N} \models_{v_1} A.r \leftarrow B.s \\
\mathbb{N} \models_{v_2} B.s \leftarrow C \\
\mathbb{N} \models_{v_1 \cap v_2} A.r \leftarrow C
\]

\[
\mathbb{N} \models_{v_1} A.r \leftarrow B.s.t \\
\mathbb{N} \models_{v_2} B.s \leftarrow C \\
\mathbb{N} \models_{v_3} D.t \leftarrow D \\
\mathbb{N} \models_{v_1 \cap v_2 \cap v_3} A.r \leftarrow D
\]

\[
\mathbb{N} \models_{v_1} A.r \leftarrow B.s \sqcap C.t \\
\mathbb{N} \models_{v_2} B.s \leftarrow D \\
\mathbb{N} \models_{v_3} C.t \leftarrow D \\
\mathbb{N} \models_{v_1 \cap v_2 \cap v_3} A.r \leftarrow D
\]
Deriving time validity

The same credential can be inferred in different ways, with different time validity; the following rule takes into account this possibility:

\[ \mathcal{N} \vdash_{\tau} C \quad \mathcal{N} \vdash_{\tau'} C \quad \mathcal{N} \vdash_{\tau \cup \tau'} C \]

If such a rule is used whenever possible throughout the inference of \( \mathcal{N} \vdash_{\tau} C \), then we can prove that

\[ \mathcal{N} \vdash_{\tau} C \text{ if and only if } \tau \in \psi \text{ and } \mathcal{N} \text{ has a semantics at time } \tau. \]
The same credential can be inferred in different ways, with different time validity; the following rule takes into account this possibility:

\[
\begin{align*}
\mathbb{N} \models_{\nu_1} C & \quad \mathbb{N} \models_{\nu_2} C \\
\mathbb{N} \models_{\nu_1 \cup \nu_2} C
\end{align*}
\]

If such a rule is used whenever possible throughout the inference of \(\mathbb{N} \models_{\nu} C\), then we can prove that

\[\mathbb{N} \models_{\tau} C \text{ if and only if } \tau \in \nu \text{ and } \mathbb{N} \text{ has a semantics at time } \tau.\]
Characterise necessary and conflicting context credentials for an inference to hold.

We aim at an inference system with judgements of the form

$$\mathcal{N} \vdash_{\tau} \phi \ c$$

meaning that \(c\) is derivable from \(\mathcal{N}\) at time \(\tau\) in any execution context that satisfies \(\phi\).

\(\phi\) is a propositional formula over the atoms \(B \in A. r\), i.e.

$$\phi ::= \top | B \in A. r | \neg \phi | \phi_1 \land \phi_2 | \phi_1 \lor \phi_2$$
Deriving Environmental Knowledge

Characterise necessary and conflicting context credentials for an inference to hold.

We aim at an inference system with judgements of the form

\[ \text{ℕ} \vdash_τ^φ \text{c} \]

meaning that \( \text{c} \) is derivable from \( \text{ℕ} \) at time \( τ \) in any execution context that satisfies \( φ \).

\( φ \) is a propositional formula over the atoms \( B \in A.r \), i.e.

\[ \phi ::= \top | B \in A.r | \neg φ | φ_1 \land φ_2 | φ_1 \lor φ_2 \]
Such propositional formulae characterise sets of CDCs:

**Definition**

\[ \mathbb{N} \models_\tau \text{tt} \iff \mathbb{N} \text{ has a semantics at time } \tau \]

\[ \mathbb{N} \models_\tau B \in A.r \iff B \in \llbracket \mathbb{N} \rrbracket_\tau(A.r) \]

\[ \mathbb{N} \models_\tau \neg \phi \iff \mathbb{N} \not\models_\tau \phi \]

\[ \mathbb{N} \models_\tau \phi_1 \land \phi_2 \iff \mathbb{N} \models_\tau \phi_1 \text{ and } \mathbb{N} \models_\tau \phi_2 \]

\[ \mathbb{N} \models_\tau \phi_1 \lor \phi_2 \iff \mathbb{N} \models_\tau \phi_1 \text{ or } \mathbb{N} \models_\tau \phi_2 \]
Straightforward adaptations of the previous rules:

\[
\text{if } \bigwedge_i B_i \in A_i.r_i \land \bigwedge_j B'_i \notin A'_j.r'_j \text{ then } c \text{ in } \nu \in \mathbb{N} \\
\tau \in \nu \quad \forall i \cdot \mathbb{N} \vdash_{\tau} \phi_i \quad A_i.r_i \leftarrow B_i
\]

\[
\mathbb{N} \vdash_{\tau} \bigwedge \phi_i \land \bigwedge B'_i \notin A'_j.r'_j \\
C
\]

\[
\mathbb{N} \vdash_{\tau} \phi_1 \quad A.r \leftarrow B.s 
\mathbb{N} \vdash_{\tau} \phi_2 \quad B.s \leftarrow C
\]

\[
\mathbb{N} \vdash_{\tau} \phi_1 \quad A.r \leftarrow C
\]

\[
\mathbb{N} \vdash_{\tau} \phi_1 \quad A.r \leftarrow B.s.t 
\mathbb{N} \vdash_{\tau} \phi_2 \quad B.s \leftarrow C 
\mathbb{N} \vdash_{\tau} \phi_3 \quad C.t \leftarrow D
\]

\[
\mathbb{N} \vdash_{\tau} \phi_1 \land \phi_2 \land \phi_3 \quad A.r \leftarrow D
\]

\[
\mathbb{N} \vdash_{\tau} \phi_1 \quad A.r \leftarrow B.s \sqcap C.t 
\mathbb{N} \vdash_{\tau} \phi_2 \quad B.s \leftarrow D 
\mathbb{N} \vdash_{\tau} \phi_3 \quad C.t \leftarrow D
\]

\[
\mathbb{N} \vdash_{\tau} \phi_1 \land \phi_2 \land \phi_3 \quad A.r \leftarrow D
\]
Deriving Environmental Knowledge

Straightforward adaptations of the previous rules:

\[
\text{if } \bigwedge_i B_i \in A_i.r_i \land \bigwedge_j B'_j \notin A'_j.r'_j \text{ then } c \in v \in \mathbb{N}
\]

\[
\tau \in v \quad \forall i \cdot \mathbb{N} \models \phi_i A_i.r_i \leftarrow B_i
\]

\[
\mathbb{N} \models \bigwedge_i \phi_i \land \bigwedge_j B'_j \notin A'_j.r'_j \quad c
\]

\[
\mathbb{N} \models \phi_1 A.r \leftarrow B.s \quad \mathbb{N} \models \phi_2 B.s \leftarrow C
\]

\[
\mathbb{N} \models \phi_1 \land \phi_2 A.r \leftarrow C
\]

\[
\mathbb{N} \models \phi_1 A.r \leftarrow B.s.t \quad \mathbb{N} \models \phi_2 B.s \leftarrow C \quad \mathbb{N} \models \phi_3 C.t \leftarrow D
\]

\[
\mathbb{N} \models \phi_1 \land \phi_2 \land \phi_3 A.r \leftarrow D
\]

\[
\mathbb{N} \models \phi_1 A.r \leftarrow B.s \sqcap C.t \quad \mathbb{N} \models \phi_2 B.s \leftarrow D \quad \mathbb{N} \models \phi_3 C.t \leftarrow D
\]

\[
\mathbb{N} \models \phi_1 \land \phi_2 \land \phi_3 A.r \leftarrow D
\]
Deriving Environmental Knowledge (4)

A rule like

\[
\begin{align*}
\text{N} \models \phi_1 \quad \text{N} \models \phi_2 \\
\therefore \text{N} \models \phi_1 \lor \phi_2
\end{align*}
\]

is sound, but not strictly necessary.

An additional set of axioms is needed for the inference system to work properly:

\[
\begin{align*}
\text{N} \models B \in A \quad r \\
\therefore \text{N} \models A. r \leftarrow B
\end{align*}
\]

Theorem (soundness and completeness)

Let \( \mathcal{N} \) be such that \( \mathcal{N} \cup \mathcal{N}' \models_\tau \phi \); then, \( \mathcal{N} \models_\tau \phi \) \iff \( \mathcal{N} \cup \mathcal{N}' \models_\tau \phi \).
Deriving Environmental Knowledge

A rule like

\[ \models \phi_1 \quad \models \phi_2 \]

is sound, but not strictly necessary.

An additional set of axioms is needed for the inference system work properly:

\[ \models B \in A.r \quad A.r \leftarrow B \]

**Theorem (soundness and completeness)**

Let \( \mathcal{N} \) be such that \( \mathcal{N} \cup \mathcal{N}' \models \phi \); then, \( \mathcal{N} \models \phi \quad \text{iff} \quad \mathcal{N} \cup \mathcal{N}' \models \phi \).
Deriving Environmental Knowledge

A rule like

\[
\begin{align*}
\mathbb{N} \models \phi_1 \quad \mathbb{N} \models \phi_2 \\
\hline
\mathbb{N} \models \phi_1 \lor \phi_2
\end{align*}
\]

is sound, but not strictly necessary.

An additional set of axioms is needed for the inference system work properly:

\[
\mathbb{N} \models B \in A.r \\
\hline
A.r \leftarrow B
\]

Theorem (soundness and completeness)

Let \( \mathbb{N}' \) be such that \( \mathbb{N} \cup \mathbb{N}' \models \phi \); then, \( \mathbb{N} \vdash_{\tau} \phi \) iff \( \mathbb{N} \cup \mathbb{N}' \vdash_{\tau} \phi \).
Conclusion

- Expressive variant of $RT_0$ with enhanced inference system;
- Set-theoretic and logic-programming semantics for CDCs;
- Use of stable model theory to handle divergence arising from the presence of negative premises;
- Inference of constraints on the execution environment; these are equivalent to abductive constraint LP (cf. the paper)

Future Work

- Allow CDCs with richer kinds of premises; e.g.,
  
  $\text{if } A.r \subseteq B.s \text{ then } c \text{ in } v \quad \text{or} \quad \text{if } A.r \cap B.s = \emptyset \text{ then } c \text{ in } v$

- Allow negative forms of delegations; e.g.,
  
  $A.r \leftarrow B.s \cap \neg C.t$
Conclusion

- Expressive variant of RT\(_0\) with enhanced inference system;
- Set-theoretic and logic-programming semantics for CDCs;
- Use of stable model theory to handle divergence arising from the presence of negative premises;
- Inference of constraints on the execution environment; these are equivalent to abductive constraint LP (cf. the paper)

Future Work

- Allow CDCs with richer kinds of premises; e.g.,
  
  \[
  \text{if } A.r \subseteq B.s \text{ then } c \text{ in } \upsilon \quad \text{or} \quad \text{if } A.r \cap B.s = \emptyset \text{ then } c \text{ in } \upsilon
  \]

- Allow negative forms of delegations; e.g.,
  
  \[
  A.r \leftarrow B.s \cap \neg C.t
  \]