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. RT_0 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.

RT_0, by example

- An auditor can inspect an enterprise **ENT** only if is authorised by the UK government: **ENT.AUDITOR** ← **UK.AUDITOR**;
- An audithor is authorised if is a member of a government recognised society: **UK.AUDITOR** ← **UK.AUTHSOC.MEMBER**;
- Auditing societies must be legally registered and 'fair':

  UK.AUTHSOC ← **UK.LEGALSOC** ⊓ **UK.FAIRSOC**.

- 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**.

RT_0, by example

- An auditor can inspect an enterprise **ENT** only if is authorised by the UK government: **ENT.AUDITOR** ← **UK.AUDITOR**;
- An audithor is authorised if is a member of a government recognised society: **UK.AUDITOR** ← **UK.AUTHSOC.MEMBER**;
- Auditing societies must be legally registered and 'fair':

  UK.AUTHSOC ← **UK.LEGALSOC** ⊓ **UK.FAIRSOC**.

- 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**.

RT_0, by example

- An auditor can inspect an enterprise **Ent** only if is authorised by the UK government: **Ent.auditor** ← **UK.auditor**;
- An audithor is authorised if is a member of a government recognised society: **UK.auditor** ← **UK.authSoc.member**;
- Auditing societies must be legally registered and 'fair':

UK.authSoc ← **UK.legalSoc** ⊓ **UK.fairSoc**.

- 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**.

RT_0, by example

- An auditor can inspect an enterprise **Ent** only if is authorised by the UK government: **Ent.auditor** ← **UK.auditor**;
- An audithor is authorised if is a member of a government recognised society: **UK.auditor** ← **UK.authSoc.member**;
- Auditing societies must be legally registered and `fair':

  UK.authSoc ← **UK.legalSoc** ⊓ **UK.fairSoc**.

- 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**.

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.

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.

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.

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.

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.

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.

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.

$$\frac{c \in P}{P \succ c} \qquad \frac{P \succ A.r \leftarrow B.s \qquad P \succ B.s \leftarrow C}{P \succ A.r \leftarrow C}$$

$$\frac{P \succ A.r \leftarrow B.s.t \qquad P \succ B.s \leftarrow C \qquad P \succ C.t \leftarrow D}{P \succ A.r \leftarrow D}$$

$$\frac{P \succ A.r \leftarrow B.s \sqcap C.t \qquad P \succ B.s \leftarrow D \qquad P \succ C.t \leftarrow D}{P \succ A.r \leftarrow D}$$

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.

$$\frac{c \in P}{P \succ c} \qquad \frac{P \succ A.r \leftarrow B.s \qquad P \succ B.s \leftarrow C}{P \succ A.r \leftarrow C}$$

$$\frac{P \succ A.r \leftarrow B.s.t \qquad P \succ B.s \leftarrow C \qquad P \succ C.t \leftarrow D}{P \succ A.r \leftarrow D}$$

$$\frac{P \succ A.r \leftarrow B.s \sqcap C.t \qquad P \succ B.s \leftarrow D \qquad P \succ C.t \leftarrow D}{P \succ A.r \leftarrow D}$$

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.

$$\frac{c \in P}{P \succ c} \qquad \frac{P \succ A.r \leftarrow B.s \qquad P \succ B.s \leftarrow C}{P \succ A.r \leftarrow C}$$

$$\frac{P \succ A.r \leftarrow B.s.t \qquad P \succ B.s \leftarrow C \qquad P \succ C.t \leftarrow D}{P \succ A.r \leftarrow D}$$

$$\frac{P \succ A.r \leftarrow B.s \sqcap C.t \qquad P \succ B.s \leftarrow D \qquad P \succ C.t \leftarrow D}{P \succ A.r \leftarrow D}$$

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.

$$\frac{c \in P}{P \succ c} \qquad \frac{P \succ A.r \leftarrow B.s \qquad P \succ B.s \leftarrow C}{P \succ A.r \leftarrow C}$$

$$\frac{P \succ A.r \leftarrow B.s.t \qquad P \succ B.s \leftarrow C \qquad P \succ C.t \leftarrow D}{P \succ A.r \leftarrow D}$$

$$\frac{P \succ A.r \leftarrow B.s \sqcap C.t \qquad P \succ B.s \leftarrow D \qquad P \succ C.t \leftarrow D}{P \succ A.r \leftarrow D}$$

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.

$$\frac{c \in P}{P \succ c} \qquad \frac{P \succ A.r \leftarrow B.s \qquad P \succ B.s \leftarrow C}{P \succ A.r \leftarrow C}$$

$$\frac{P \succ A.r \leftarrow B.s.t \qquad P \succ B.s \leftarrow C \qquad P \succ C.t \leftarrow D}{P \succ A.r \leftarrow D}$$

$$\frac{P \succ A.r \leftarrow B.s \sqcap C.t \qquad P \succ B.s \leftarrow D \qquad P \succ C.t \leftarrow D}{P \succ A.r \leftarrow D}$$

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.

$$\frac{c \in P}{P \succ c} \qquad \frac{P \succ A.r \leftarrow B.s \qquad P \succ B.s \leftarrow C}{P \succ A.r \leftarrow C}$$

$$\frac{P \succ A.r \leftarrow B.s.t \qquad P \succ B.s \leftarrow C \qquad P \succ C.t \leftarrow D}{P \succ A.r \leftarrow D}$$

$$\frac{P \succ A.r \leftarrow B.s \sqcap C.t \qquad P \succ B.s \leftarrow D \qquad P \succ C.t \leftarrow D}{P \succ A.r \leftarrow D}$$

The auditing example, formalised

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

$$\dfrac{\dfrac{P \succ \text{UK.LEGALSOC} \leftarrow \text{BS} \qquad P \succ \text{UK.FAIRSOC} \leftarrow \text{BS} \qquad P \succ \text{UK.AUTHSOC} \leftarrow \text{UK.LEGALSOC} \sqcap \text{UK.FAIRSOC}}{P \succ \text{UK.AUTHSOC} \leftarrow \text{BS}} \qquad P \succ \text{BS.MEMBER} \leftarrow \text{B} \qquad P \succ \text{UK.AUDITOR} \leftarrow \text{UK.AUTHSOC.MEMBER}}{P \succ \text{UK.AUDITOR} \leftarrow \text{B}}$$

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

$$\dfrac{P \succ \text{ENT.AUDITOR} \leftarrow \text{UK.AUDITOR} \qquad P \succ \text{UK.AUDITOR} \leftarrow \text{B}}{P \succ \text{ENT.AUDITOR} \leftarrow \text{B}}$$

The auditing example, formalised

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

$$\frac{P \succ \text{UK.LEGALSOC} \leftarrow \text{BS} \qquad P \succ \text{UK.FAIRSOC} \leftarrow \text{BS} \qquad P \succ \text{UK.AUTHSOC} \leftarrow \text{UK.LEGALSOC} \sqcap \text{UK.FAIRSOC}}{P \succ \text{UK.AUTHSOC} \leftarrow \text{BS}}$$

$$\frac{P \succ \text{BS.MEMBER} \leftarrow \text{B} \qquad P \succ \text{UK.AUDITOR} \leftarrow \text{UK.AUTHSOC.MEMBER}}{P \succ \text{UK.AUDITOR} \leftarrow \text{B}}$$

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

$$\frac{P \succ \text{ENT.AUDITOR} \leftarrow \text{UK.AUDITOR} \qquad P \succ \text{UK.AUDITOR} \leftarrow \text{B}}{P \succ \text{ENT.AUDITOR} \leftarrow \text{B}}$$

The auditing example, formalised

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

$$\dfrac{\dfrac{P \succ \mathbf{UK.LegalSoc} \leftarrow \mathbf{BS} \qquad P \succ \mathbf{UK.FairSoc} \leftarrow \mathbf{BS} \qquad P \succ \mathbf{UK.AuthSoc} \leftarrow \mathbf{UK.LegalSoc} \sqcap \mathbf{UK.FairSoc}}{P \succ \mathbf{UK.AuthSoc} \leftarrow \mathbf{BS}} \qquad P \succ \mathbf{BS.member} \leftarrow \mathbf{B} \qquad P \succ \mathbf{UK.auditor} \leftarrow \mathbf{UK.AuthSoc.member}}{P \succ \mathbf{UK.auditor} \leftarrow \mathbf{B}}$$

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

$$\dfrac{P \succ \mathbf{Ent.auditor} \leftarrow \mathbf{UK.auditor} \qquad P \succ \mathbf{UK.auditor} \leftarrow \mathbf{B}}{P \succ \mathbf{Ent.auditor} \leftarrow \mathbf{B}}$$

The auditing example, formalised

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

$$\dfrac{\dfrac{P \succ \mathbf{UK.LEGALSOC} \leftarrow \mathbf{BS} \qquad P \succ \mathbf{UK.FAIRSOC} \leftarrow \mathbf{BS} \qquad P \succ \mathbf{UK.AUTHSOC} \leftarrow \mathbf{UK.LEGALSOC} \sqcap \mathbf{UK.FAIRSOC}}{P \succ \mathbf{UK.AUTHSOC} \leftarrow \mathbf{BS}} \qquad P \succ \mathbf{BS.MEMBER} \leftarrow \mathbf{B} \qquad P \succ \mathbf{UK.AUDITOR} \leftarrow \mathbf{UK.AUTHSOC.MEMBER}}{P \succ \mathbf{UK.AUDITOR} \leftarrow \mathbf{B}}$$

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

$$\dfrac{P \succ \mathbf{ENT.AUDITOR} \leftarrow \mathbf{UK.AUDITOR} \qquad P \succ \mathbf{UK.AUDITOR} \leftarrow \mathbf{B}}{P \succ \mathbf{ENT.AUDITOR} \leftarrow \mathbf{B}}$$

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 τ:

  $$\text{UK.LEGALSOC} \leftarrow \text{BSOC in } [\tau, +\infty)$$

- 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:

  $$\text{UK.FAIRSOC} \leftarrow \text{BSOC in } v_1 \quad , \quad \text{BSOC.MEMBER} \leftarrow \text{B in } v_2$$

- **B** can inspect **ENT** if he is authorised and is not one of **ENT**'s employees:

  $$\text{if B} \in \text{UK.AUDITOR} \wedge \text{B} \notin \text{ENT.EMPLOYEE then ENT.AUDITOR} \leftarrow \text{B}$$

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 τ:

$$\textbf{UK.legalSoc} \leftarrow \textbf{BSoc in } [\tau, +\infty)$$

- UK's fairness certificates are valid only for a period of time υ_1, and **B** is a member of **BSoc** for a fixed period υ_2:

$$\textbf{UK.fairSoc} \leftarrow \textbf{BSoc in } \upsilon_1 \quad , \quad \textbf{BSoc.member} \leftarrow \textbf{B in } \upsilon_2$$

- **B** can inspect **Ent** if he is authorised and is not one of **Ent**'s employees:

$$\textbf{if B} \in \textbf{UK.auditor} \wedge \textbf{B} \notin \textbf{Ent.employee} \ \ \textbf{then Ent.auditor} \leftarrow \textbf{B}$$

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 τ:

$$\textbf{UK.LEGALSOC} \leftarrow \textbf{BSOC in } [\tau, +\infty)$$

- 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:

$$\textbf{UK.FAIRSOC} \leftarrow \textbf{BSOC in } v_1 \quad , \qquad \textbf{BSOC.MEMBER} \leftarrow \textbf{B in } v_2$$

- **B** can inspect **Ent** if he is authorised and is not one of **Ent**'s employees:

$$\textbf{if B} \in \textbf{UK.AUDITOR} \wedge \textbf{B} \notin \textbf{ENT.EMPLOYEE} \;\; \textbf{then ENT.AUDITOR} \leftarrow \textbf{B}$$

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: $g ::= \mathbf{tt} \mid B \in A.r \mid B \notin A.r \mid g_1 \wedge g_2$

TIME VALIDITY: $v ::= [\tau_1, \tau_2] \mid [\tau_1, \tau_2) \mid (\tau_1, \tau_2] \mid (\tau_1, \tau_2)$
$\mid (-\infty, \tau] \mid (-\infty, \tau) \mid [\tau, +\infty) \mid (\tau, +\infty)$
$\mid (-\infty, +\infty) \mid v_1 \cup v_2 \mid v_1 \cap v_2 \mid v_1 \setminus v_2$

CDCS: $\chi ::= \mathbf{if}\ g\ \mathbf{then}\ c\ \mathbf{in}\ v$

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: $g ::= \mathbf{tt} \mid B \in A.r \mid B \notin A.r \mid g_1 \land g_2$

TIME VALIDITY: $v ::= [\tau_1, \tau_2] \mid [\tau_1, \tau_2) \mid (\tau_1, \tau_2] \mid (\tau_1, \tau_2)$
$\mid (-\infty, \tau] \mid (-\infty, \tau) \mid [\tau, +\infty) \mid (\tau, +\infty)$
$\mid (-\infty, +\infty) \mid v_1 \cup v_2 \mid v_1 \cap v_2 \mid v_1 \setminus v_2$

CDCS: $\chi ::= \textbf{if } g \textbf{ then } c \textbf{ in } v$

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: $g ::= \mathbf{tt} \mid B \in A.r \mid B \notin A.r \mid g_1 \wedge g_2$

TIME VALIDITY: $v ::= [\tau_1, \tau_2] \mid [\tau_1, \tau_2) \mid (\tau_1, \tau_2] \mid (\tau_1, \tau_2) \mid (-\infty, \tau] \mid (-\infty, \tau) \mid [\tau, +\infty) \mid (\tau, +\infty) \mid (-\infty, +\infty) \mid v_1 \cup v_2 \mid v_1 \cap v_2 \mid v_1 \setminus v_2$

CDCS: $\chi ::= \textbf{if } g \textbf{ then } c \textbf{ in } v$

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: $g ::= \mathbf{tt} \mid B \in A.r \mid B \notin A.r \mid g_1 \wedge g_2$

TIME VALIDITY: $v ::= [\tau_1, \tau_2] \mid [\tau_1, \tau_2) \mid (\tau_1, \tau_2] \mid (\tau_1, \tau_2)$
$\mid (-\infty, \tau] \mid (-\infty, \tau) \mid [\tau, +\infty) \mid (\tau, +\infty)$
$\mid (-\infty, +\infty) \mid v_1 \cup v_2 \mid v_1 \cap v_2 \mid v_1 \setminus v_2$

CDCS: $\chi ::= \textbf{if } g \textbf{ then } c \textbf{ in } v$

An inference system for CDCs (1)

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

Judgements take the form

$$\aleph \vdash_{\tau} c$$

and mean that c can be inferred, at time τ, from $\aleph$.

This entails that $\aleph$ satisfies

- all the positive guards of the CDCs used in the inference;
- none of their negative guards.

An inference system for CDCs (2)

The key rule is:

Rules

$$\textbf{if } \bigwedge_i B_i \in A_i.r_i \wedge \bigwedge_j B'_j \notin A'_j.r'_j \textbf{ then } c \textbf{ in } \upsilon \in \aleph$$

$$\frac{\forall i \,.\, \aleph \vdash_\tau A_i.r_i \leftarrow B_i \qquad \forall j \,.\, \aleph \nvdash_\tau A'_j.r'_j \leftarrow B'_j \qquad \tau \in \upsilon}{\aleph \vdash_\tau c}$$

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 τ.

An inference system for CDCs (3)

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

Rules

$$\frac{\aleph \vdash_{\tau} A.r \leftarrow B.s \qquad \aleph \vdash_{\tau} B.s \leftarrow C}{\aleph \vdash_{\tau} A.r \leftarrow C}$$

$$\frac{\aleph \vdash_{\tau} A.r \leftarrow B.s.t \qquad \aleph \vdash_{\tau} B.s \leftarrow C \qquad \aleph \vdash_{\tau} C.t \leftarrow D}{\aleph \vdash_{\tau} A.r \leftarrow D}$$

$$\frac{\aleph \vdash_{\tau} A.r \leftarrow B.s \sqcap C.t \qquad \aleph \vdash_{\tau} B.s \leftarrow D \qquad \aleph \vdash_{\tau} C.t \leftarrow D}{\aleph \vdash_{\tau} A.r \leftarrow D}$$

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.**

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 (1)

Characterise the instants when a given inference holds.

if $\bigwedge_i B_i \in A_i.r_i \;\wedge\; \bigwedge_j B'_j \notin A'_j.r'_j$ **then** c **in** $\upsilon \in \aleph$

$$\frac{\forall i \,.\, \aleph \Vdash_{\upsilon_i} A_i.r_i \leftarrow B_i \qquad \forall j \,.\, \aleph \Vdash_{\upsilon_j} A'_j.r'_j \leftarrow B'_j}{\aleph \Vdash_{(\upsilon \,\cap\, \cap_i \upsilon_i) \setminus \cup_j \upsilon_j} c}$$

$$\frac{\aleph \Vdash_{\upsilon_1} A.r \leftarrow B.s \qquad \aleph \Vdash_{\upsilon_2} B.s \leftarrow C}{\aleph \Vdash_{\upsilon_1 \cap \upsilon_2} A.r \leftarrow C}$$

$$\frac{\aleph \Vdash_{\upsilon_1} A.r \leftarrow B.s.t \qquad \aleph \Vdash_{\upsilon_2} B.s \leftarrow C \qquad \aleph \Vdash_{\upsilon_3} D.t \leftarrow D}{\aleph \Vdash_{\upsilon_1 \cap \upsilon_2 \cap \upsilon_3} A.r \leftarrow D}$$

$$\frac{\aleph \Vdash_{\upsilon_1} A.r \leftarrow B.s \sqcap C.t \qquad \aleph \Vdash_{\upsilon_2} B.s \leftarrow D \qquad \aleph \Vdash_{\upsilon_3} C.t \leftarrow D}{\aleph \Vdash_{\upsilon_1 \cap \upsilon_2 \cap \upsilon_3} A.r \leftarrow D}$$

Deriving time validity (1)

Characterise the instants when a given inference holds.

$$\textbf{if } \bigwedge_i B_i \in A_i.r_i \;\wedge\; \bigwedge_j B'_j \notin A'_j.r'_j \textbf{ then } c \textbf{ in } \upsilon \in \aleph$$

$$\frac{\forall i \,.\, \aleph \Vdash_{\upsilon_i} A_i.r_i \leftarrow B_i \qquad \forall j \,.\, \aleph \Vdash_{\upsilon_j} A'_j.r'_j \leftarrow B'_j}{\aleph \Vdash_{(\upsilon \,\cap\, \cap_i \upsilon_i) \setminus \cup_j \upsilon_j} c}$$

$$\frac{\aleph \Vdash_{\upsilon_1} A.r \leftarrow B.s \qquad \aleph \Vdash_{\upsilon_2} B.s \leftarrow C}{\aleph \Vdash_{\upsilon_1 \cap \upsilon_2} A.r \leftarrow C}$$

$$\frac{\aleph \Vdash_{\upsilon_1} A.r \leftarrow B.s.t \qquad \aleph \Vdash_{\upsilon_2} B.s \leftarrow C \qquad \aleph \Vdash_{\upsilon_3} D.t \leftarrow D}{\aleph \Vdash_{\upsilon_1 \cap \upsilon_2 \cap \upsilon_3} A.r \leftarrow D}$$

$$\frac{\aleph \Vdash_{\upsilon_1} A.r \leftarrow B.s \sqcap C.t \qquad \aleph \Vdash_{\upsilon_2} B.s \leftarrow D \qquad \aleph \Vdash_{\upsilon_3} C.t \leftarrow D}{\aleph \Vdash_{\upsilon_1 \cap \upsilon_2 \cap \upsilon_3} A.r \leftarrow D}$$

Deriving time validity (2)

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

$$\frac{\aleph \Vdash_{\upsilon_1} c \qquad \aleph \Vdash_{\upsilon_2} c}{\aleph \Vdash_{\upsilon_1 \cup \upsilon_2} c}$$

If such a rule is used whenever possible throughout the inference of $\aleph \Vdash_{\upsilon} c$, then we can prove that

$\aleph \vdash_{\tau} c$ **if and only if** $\tau \in \upsilon$ **and** $\aleph$ **has a semantics at time** τ**.**

Deriving time validity (2)

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

$$\frac{\aleph \Vdash_{\upsilon_1} c \qquad \aleph \Vdash_{\upsilon_2} c}{\aleph \Vdash_{\upsilon_1 \cup \upsilon_2} c}$$

If such a rule is used whenever possible throughout the inference of $\aleph \Vdash_{\upsilon} c$, then we can prove that

$\aleph \vdash_{\tau} c$ if and only if $\tau \in \upsilon$ and $\aleph$ has a semantics at time τ.

Deriving Environmental Knowledge (1)

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

We aim at an inference system with judgements of the form

$$\aleph \Vdash^{\phi}_{\tau} c$$

meaning that c is derivable from $\aleph$ at time τ in any execution context that satisfies ϕ.

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

$$\phi \ ::= \ \mathbf{tt} \mid B \in A.r \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2$$

Deriving Environmental Knowledge (1)

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

We aim at an inference system with judgements of the form

$$\aleph \Vdash^{\phi}_{\tau} c$$

meaning that c is derivable from $\aleph$ at time τ in any execution context that satisfies ϕ.

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

$$\phi ::= \mathbf{tt} \mid B \in A.r \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2$$

Deriving Environmental Knowledge (2)

Such propositional formulae characterise sets of CDCs:

Definition

$$\aleph \models_\tau \mathbf{tt} \quad \text{iff} \quad \aleph \text{ has a semantics at time } \tau$$

$$\aleph \models_\tau B \in A.r \quad \text{iff} \quad B \in [\![\aleph]\!]_\tau(A.r)$$

$$\aleph \models_\tau \neg\phi \quad \text{iff} \quad \aleph \not\models_\tau \phi$$

$$\aleph \models_\tau \phi_1 \wedge \phi_2 \quad \text{iff} \quad \aleph \models_\tau \phi_1 \quad \text{and} \quad \aleph \models_\tau \phi_2$$

$$\aleph \models_\tau \phi_1 \vee \phi_2 \quad \text{iff} \quad \aleph \models_\tau \phi_1 \quad \text{or} \quad \aleph \models_\tau \phi_2$$

Deriving Environmental Knowledge (3)

Straightforward adaptions of the previous rules:

$$\textbf{if } \bigwedge_i B_i \in A_i.r_i \ \wedge\ \bigwedge_j B'_j \notin A'_j.r'_j \textbf{ then } c \textbf{ in } \upsilon \in \aleph$$

$$\frac{\tau \in \upsilon \qquad \forall i\,.\ \aleph \Vdash_\tau^{\phi_i} A_i.r_i \leftarrow B_i}{\aleph \Vdash_\tau^{\bigwedge_i \phi_i \ \wedge\ \bigwedge_j B'_j \notin A'_j.r'_j} c}$$

$$\frac{\aleph \Vdash_\tau^{\phi_1} A.r \leftarrow B.s \qquad \aleph \Vdash_\tau^{\phi_2} B.s \leftarrow C}{\aleph \Vdash_\tau^{\phi_1 \wedge \phi_2} A.r \leftarrow C}$$

$$\frac{\aleph \Vdash_\tau^{\phi_1} A.r \leftarrow B.s.t \qquad \aleph \Vdash_\tau^{\phi_2} B.s \leftarrow C \qquad \aleph \Vdash_\tau^{\phi_3} C.t \leftarrow D}{\aleph \Vdash_\tau^{\phi_1 \wedge \phi_2 \wedge \phi_3} A.r \leftarrow D}$$

$$\frac{\aleph \Vdash_\tau^{\phi_1} A.r \leftarrow B.s \sqcap C.t \qquad \aleph \Vdash_\tau^{\phi_2} B.s \leftarrow D \qquad \aleph \Vdash_\tau^{\phi_3} C.t \leftarrow D}{\aleph \Vdash_\tau^{\phi_1 \wedge \phi_2 \wedge \phi_3} A.r \leftarrow D}$$

Deriving Environmental Knowledge (3)

Straightforward adaptions of the previous rules:

$$\textbf{if } \bigwedge_i B_i \in A_i.r_i \ \wedge\ \bigwedge_j B'_j \notin A'_j.r'_j \textbf{ then } c \textbf{ in } \upsilon \in \aleph$$

$$\frac{\tau \in \upsilon \qquad \forall i \,.\, \aleph \Vdash^{\phi_i}_{\tau} A_i.r_i \leftarrow B_i}{\aleph \Vdash^{\bigwedge_i \phi_i \ \wedge\ \bigwedge_j B'_j \notin A'_j.r'_j}_{\tau} c}$$

$$\frac{\aleph \Vdash^{\phi_1}_{\tau} A.r \leftarrow B.s \qquad \aleph \Vdash^{\phi_2}_{\tau} B.s \leftarrow C}{\aleph \Vdash^{\phi_1 \wedge \phi_2}_{\tau} A.r \leftarrow C}$$

$$\frac{\aleph \Vdash^{\phi_1}_{\tau} A.r \leftarrow B.s.t \qquad \aleph \Vdash^{\phi_2}_{\tau} B.s \leftarrow C \qquad \aleph \Vdash^{\phi_3}_{\tau} C.t \leftarrow D}{\aleph \Vdash^{\phi_1 \wedge \phi_2 \wedge \phi_3}_{\tau} A.r \leftarrow D}$$

$$\frac{\aleph \Vdash^{\phi_1}_{\tau} A.r \leftarrow B.s \sqcap C.t \qquad \aleph \Vdash^{\phi_2}_{\tau} B.s \leftarrow D \qquad \aleph \Vdash^{\phi_3}_{\tau} C.t \leftarrow D}{\aleph \Vdash^{\phi_1 \wedge \phi_2 \wedge \phi_3}_{\tau} A.r \leftarrow D}$$

Deriving Environmental Knowledge (4)

A rule like

$$\frac{\aleph \Vdash_{\mathcal{T}}^{\phi_1} c \qquad \aleph \Vdash_{\mathcal{T}}^{\phi_2} c}{\aleph \Vdash_{\mathcal{T}}^{\phi_1 \vee \phi_2} c}$$

is sound, but not strictly necessary.

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

$$\frac{}{\aleph \Vdash_{\mathcal{T}}^{B \in A.r} A.r \leftarrow B}$$

Theorem (soundness and completeness)

Let $\aleph'$ be such that $\aleph \cup \aleph' \vDash_{\mathcal{T}} \phi$; then, $\aleph \Vdash_{\mathcal{T}}^{\phi} c$ iff $\aleph \cup \aleph' \vdash_{\mathcal{T}} c$.

Deriving Environmental Knowledge (4)

A rule like

$$\frac{\aleph \ \Vdash_{\mathcal{T}}^{\phi_1} c \qquad \aleph \ \Vdash_{\mathcal{T}}^{\phi_2} c}{\aleph \ \Vdash_{\mathcal{T}}^{\phi_1 \vee \phi_2} c}$$

is sound, but not strictly necessary.

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

$$\frac{}{\aleph \ \Vdash_{\mathcal{T}}^{B \in A.r} A.r \leftarrow B}$$

Theorem (soundness and completeness)

Let $\aleph'$ be such that $\aleph \cup \aleph' \vDash_{\mathcal{T}} \phi$; then, $\aleph \ \Vdash_{\mathcal{T}}^{\phi} c$ iff $\aleph \cup \aleph' \vdash_{\mathcal{T}} c$.

Deriving Environmental Knowledge (4)

A rule like

$$\frac{\aleph \ \Vdash_{\mathcal{T}}^{\phi_1} c \qquad \aleph \ \Vdash_{\mathcal{T}}^{\phi_2} c}{\aleph \ \Vdash_{\mathcal{T}}^{\phi_1 \vee \phi_2} c}$$

is sound, but not strictly necessary.

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

$$\frac{}{\aleph \ \Vdash_{\mathcal{T}}^{B \in A.r} A.r \leftarrow B}$$

Theorem (soundness and completeness)

Let $\aleph'$ be such that $\aleph \cup \aleph' \vDash_{\mathcal{T}} \phi$; then, $\aleph \ \Vdash_{\mathcal{T}}^{\phi} c$ iff $\aleph \cup \aleph' \vdash_{\mathcal{T}} c$.

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.,

$$\textbf{if } A.r \subseteq B.s \textbf{ then } c \textbf{ in } \upsilon \quad \text{or} \quad \textbf{if } A.r \cap B.s = \emptyset \textbf{ then } c \textbf{ in } \upsilon$$

- Allow negative forms of delegations; e.g.,

$$A.r \leftarrow B.s \sqcap \neg C.t$$

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.,

$$\textbf{if } A.r \subseteq B.s \textbf{ then } c \textbf{ in } \upsilon \quad \text{or} \quad \textbf{if } A.r \cap B.s = \emptyset \textbf{ then } c \textbf{ in } \upsilon$$

- Allow negative forms of delegations; e.g.,

$$A.r \leftarrow B.s \sqcap \neg C.t$$