Nominal Confluence Tool

IJCAR 2016, Coimbra

Takahito Aoto (Niigata University)

Kentaro Kikuchi (Tohoku University)

Confluence in Term Rewriting

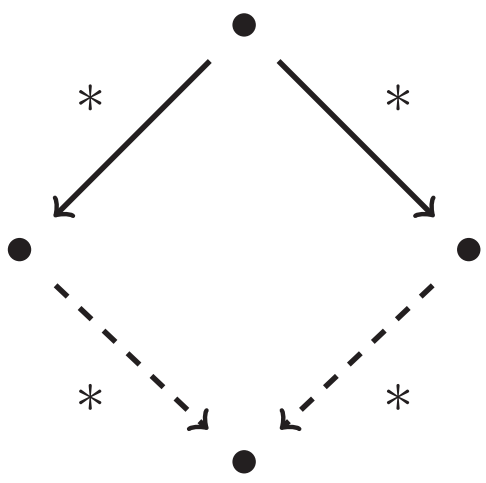
- Term rewriting is a model of computation.
- A term rewriting system (TRS) is a set of directed equations.

Confluence in Term Rewriting

- Term rewriting is a model of computation.
- A term rewriting system (TRS) is a set of directed equations.
- A TRS specifies computations in term rewriting (as rules) and equational reasoning (as axioms).
- Non-deterministic computation in term rewriting is a basis of automated theorem proving in equational logic.

Confluence in Term Rewriting

- Term rewriting is a model of computation.
- A term rewriting system (TRS) is a set of directed equations.
- A TRS specifies computations in term rewriting (as rules) and equational reasoning (as axioms).
- Non-deterministic computation in term rewriting is a basis of automated theorem proving in equational logic.

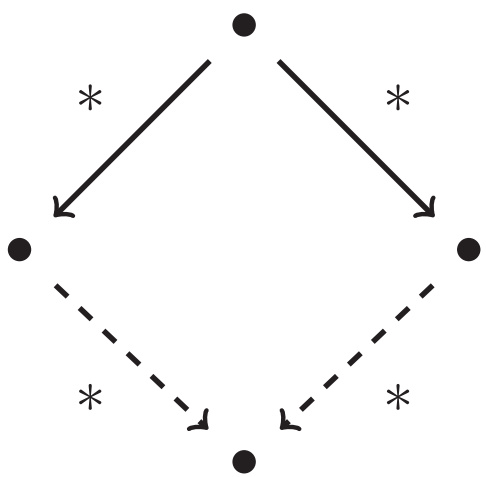


Confluence

- Confluence is a key property, which guarantees unique results of computations.

Confluence in Term Rewriting

- Term rewriting is a model of computation.
- A term rewriting system (TRS) is a set of directed equations.
- A TRS specifies computations in term rewriting (as rules) and equational reasoning (as axioms).
- Non-deterministic computation in term rewriting is a basis of automated theorem proving in equational logic.



Confluence

- Confluence is a key property, which guarantees unique results of computations.
- Numerous techniques for (dis)proving confluence have been developed.
- Motivates development of *confluence tools* (and also *confluence competition*).

Confluence Tools

Confluence Tools

Confluence tools have been developed

for various *frameworks*, various *confluence properties*.

Confluence Tools

Confluence tools have been developed

for various *frameworks*, various *confluence properties*.

- for first-order TRSs: ACP, CSI, Saigawa-CoLL, ...
- for conditional TRSs: ConCon, CO3, CoScart, ...
- for higher-order rewriting systems (HRS): ACPH, CSI^{ho}
- for ground confluence of many-sorted TRSs: AGCP

Confluence Tools

Confluence tools have been developed

for various *frameworks*, various *confluence properties*.

- for first-order TRSs: ACP, CSI, Saigawa-CoLL, ...
- for conditional TRSs: ConCon, CO3, CoScart, ...
- for higher-order rewriting systems (HRS): ACPH, CSI^{ho}
- for ground confluence of many-sorted TRSs: AGCP

This work: Confluence Tool for "*Nominal Rewriting Systems*"

Nominal Rewriting (Fernández & Gabbay, 2007)

- Extension of first-order term rewriting
- Binding mechanism
  - Nominal approach (Gabbay & Pitts, 2002)

Nominal Rewriting (Fernández & Gabbay, 2007)

- Extension of first-order term rewriting
- Binding mechanism
  - Nominal approach (Gabbay & Pitts, 2002)
  - Use variable names ($\leftrightarrow$ de Bruijn index)

Nominal Rewriting (Fernández & Gabbay, 2007)

- Extension of first-order term rewriting
- Binding mechanism
  - Nominal approach (Gabbay & Pitts, 2002)
  - Use variable names ($\leftrightarrow$ de Bruijn index)
- α-equivalence is dealt with at object-level
  - In contrast to traditional higher-order rewriting, which uses λ-calculus as meta-calculus.

Nominal Rewriting (Fernández & Gabbay, 2007)

- Extension of first-order term rewriting
- Binding mechanism
  - Nominal approach (Gabbay & Pitts, 2002)
  - Use variable names ($\leftrightarrow$ de Bruijn index)
- α-equivalence is dealt with at object-level
  - In contrast to traditional higher-order rewriting, which uses λ-calculus as meta-calculus.

Example.

$$\begin{array}{rcll} (\forall a.P) \wedge Q & \equiv & \forall a.(P \wedge Q) & (a \notin FV(Q)) \\ P \wedge (\forall a.Q) & \equiv & \forall a.(P \wedge Q) & (a \notin FV(P)) \end{array}$$

Nominal rewriting system (NRS for short)

$$\left\{ \begin{array}{rcccl} \mathtt{a}\#Q & \vdash & \mathtt{and}(\mathtt{forall}([\mathtt{a}]P), Q) & \to & \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) \\ \mathtt{a}\#P & \vdash & \mathtt{and}(P, \mathtt{forall}([\mathtt{a}]Q)) & \to & \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) \end{array} \right.$$

Nominal Rewriting (Fernández & Gabbay, 2007)

- Extension of first-order term rewriting
- Binding mechanism
  - Nominal approach (Gabbay & Pitts, 2002)
  - Use variable names ($\leftrightarrow$ de Bruijn index)
- α-equivalence is dealt with at object-level
  - In contrast to traditional higher-order rewriting, which uses λ-calculus as meta-calculus.

Example.

$$\begin{array}{lcll} (\forall a.P) \wedge Q & \equiv & \forall a.(P \wedge Q) & (a \notin FV(Q)) \\ P \wedge (\forall a.Q) & \equiv & \forall a.(P \wedge Q) & (a \notin FV(P)) \end{array}$$

Nominal rewriting system (NRS for short)

$$\left\{ \begin{array}{lcll} \mathtt{a}\#Q & \vdash & \mathtt{and}(\mathtt{forall}([\mathtt{a}]P), Q) & \rightarrow \quad \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) \\ \mathtt{a}\#P & \vdash & \mathtt{and}(P, \mathtt{forall}([\mathtt{a}]Q)) & \rightarrow \quad \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) \end{array} \right.$$

Rewriting by NRSs

$$\begin{cases} \mathtt{a}\#Q \ \vdash \ \mathtt{and}(\mathtt{forall}([\mathtt{a}]P), Q) \ \to \ \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) & (R_1) \\ \mathtt{a}\#P \ \vdash \ \mathtt{and}(P, \mathtt{forall}([\mathtt{a}]Q)) \ \to \ \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) & (R_2) \end{cases}$$

$$\mathtt{and}(\mathtt{forall}([\mathtt{a}]p(\mathtt{a})), \mathtt{forall}([\mathtt{a}]q(\mathtt{a})))$$

Rewriting by NRSs

$$\begin{cases} \mathtt{a}\#Q & \vdash & \mathtt{and}(\mathtt{forall}([\mathtt{a}]P), Q) & \to & \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) & (R_1) \\ \mathtt{a}\#P & \vdash & \mathtt{and}(P, \mathtt{forall}([\mathtt{a}]Q)) & \to & \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) & (R_2) \end{cases}$$

$$\mathtt{and}(\mathtt{forall}([\mathtt{a}]p(\mathtt{a})), \mathtt{forall}([\mathtt{a}]q(\mathtt{a})))$$

Rewriting by NRSs

$$\begin{cases} \mathtt{a}\#Q & \vdash & \mathtt{and}(\mathtt{forall}([\mathtt{a}]P), Q) & \to & \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) & (R_1) \\ \mathtt{a}\#P & \vdash & \mathtt{and}(P, \mathtt{forall}([\mathtt{a}]Q)) & \to & \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) & (R_2) \end{cases}$$

$$\begin{aligned} & \mathtt{and}(\mathtt{forall}([\mathtt{a}]p(\mathtt{a})), \mathtt{forall}([\mathtt{a}]q(\mathtt{a}))) \\ \to_{R_1} \quad & \mathtt{forall}([\mathtt{a}]\mathtt{and}(p(\mathtt{a}), \mathtt{forall}([\mathtt{a}]q(\mathtt{a})))) \end{aligned}$$

Rewriting by NRSs

$$\begin{cases} \mathtt{a}\#Q & \vdash & \mathtt{and}(\mathtt{forall}([\mathtt{a}]P), Q) & \to & \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) & (R_1) \\ \mathtt{a}\#P & \vdash & \mathtt{and}(P, \mathtt{forall}([\mathtt{a}]Q)) & \to & \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) & (R_2) \end{cases}$$

$$\begin{array}{ll} & \mathtt{and}(\mathtt{forall}([\mathtt{a}]p(\mathtt{a})), \mathtt{forall}([\mathtt{a}]q(\mathtt{a}))) \\ \to_{R_1} & \mathtt{forall}([\mathtt{a}]\mathtt{and}(p(\mathtt{a}), \mathtt{forall}([\mathtt{a}]q(\mathtt{a})))) \end{array}$$

Rewriting by NRSs

$$\begin{cases} \mathtt{a}\#Q \vdash \mathtt{and}(\mathtt{forall}([\mathtt{a}]P), Q) \to \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) & (R_1) \\ \mathtt{a}\#P \vdash \mathtt{and}(P, \mathtt{forall}([\mathtt{a}]Q)) \to \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) & (R_2) \end{cases}$$

$$\begin{aligned} & \mathtt{and}(\mathtt{forall}([\mathtt{a}]p(\mathtt{a})), \mathtt{forall}([\mathtt{a}]q(\mathtt{a}))) \\ \to_{R_1} \quad & \mathtt{forall}([\mathtt{a}]\mathtt{and}(p(\mathtt{a}), \mathtt{forall}([\mathtt{a}]q(\mathtt{a})))) \end{aligned}$$

Rewriting by NRSs

$$\begin{cases} \mathtt{a}\#Q & \vdash & \mathtt{and}(\mathtt{forall}([\mathtt{a}]P), Q) & \to & \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) & (R_1) \\ \mathtt{a}\#P & \vdash & \mathtt{and}(P, \mathtt{forall}([\mathtt{a}]Q)) & \to & \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) & (R_2) \end{cases}$$

$$\begin{aligned} & \mathtt{and}(\mathtt{forall}([\mathtt{a}]p(\mathtt{a})), \mathtt{forall}([\mathtt{a}]q(\mathtt{a}))) \\ \to_{R_1} \quad & \mathtt{forall}([\mathtt{a}]\mathtt{and}(p(\mathtt{a}), \mathtt{forall}([\mathtt{a}]q(\mathtt{a})))) \\ \approx_{\alpha} \quad & \mathtt{forall}([\mathtt{a}]\mathtt{and}(p(\mathtt{a}), \mathtt{forall}([\mathtt{b}]q(\mathtt{b}))))\end{aligned}$$

Rewriting by NRSs

$$\begin{cases} \mathtt{a}\#Q \vdash \mathtt{and}(\mathtt{forall}([\mathtt{a}]P), Q) \to \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) & (R_1) \\ \mathtt{a}\#P \vdash \mathtt{and}(P, \mathtt{forall}([\mathtt{a}]Q)) \to \mathtt{forall}([\mathtt{a}]\mathtt{and}(P, Q)) & (R_2) \end{cases}$$

$$\mathtt{b}\#P \vdash \mathtt{and}(P, \mathtt{forall}([\mathtt{b}]Q)) \to \mathtt{forall}([\mathtt{b}]\mathtt{and}(P, Q)) \quad (R_2^{(\mathtt{a}\ \mathtt{b})})$$

Permutatively renamed variant of R_2

$$\begin{aligned} & \mathtt{and}(\mathtt{forall}([\mathtt{a}]p(\mathtt{a})), \mathtt{forall}([\mathtt{a}]q(\mathtt{a}))) \\ \to_{R_1} \quad & \mathtt{forall}([\mathtt{a}]\mathtt{and}(p(\mathtt{a}), \mathtt{forall}([\mathtt{a}]q(\mathtt{a})))) \\ \approx_{\alpha} \quad & \mathtt{forall}([\mathtt{a}]\mathtt{and}(p(\mathtt{a}), \mathtt{forall}([\mathtt{b}]q(\mathtt{b})))) \end{aligned}$$

Rewriting by NRSs

$$\begin{cases} \mathsf{a}\#Q \vdash \mathtt{and}(\mathtt{forall}([\mathsf{a}]P), Q) \to \mathtt{forall}([\mathsf{a}]\mathtt{and}(P, Q)) & (R_1) \\ \mathsf{a}\#P \vdash \mathtt{and}(P, \mathtt{forall}([\mathsf{a}]Q)) \to \mathtt{forall}([\mathsf{a}]\mathtt{and}(P, Q)) & (R_2) \end{cases}$$

$$\mathsf{b}\#P \vdash \mathtt{and}(P, \mathtt{forall}([\mathsf{b}]Q)) \to \mathtt{forall}([\mathsf{b}]\mathtt{and}(P, Q)) \quad (R_2^{(\mathsf{a}\ \mathsf{b})})$$

Permutatively renamed variant of R_2

$$\begin{aligned} & \mathtt{and}(\mathtt{forall}([\mathsf{a}]p(\mathsf{a})), \mathtt{forall}([\mathsf{a}]q(\mathsf{a}))) \\ \to_{R_1} \quad & \mathtt{forall}([\mathsf{a}]\mathtt{and}(p(\mathsf{a}), \mathtt{forall}([\mathsf{a}]q(\mathsf{a})))) \\ \approx_\alpha \quad & \mathtt{forall}([\mathsf{a}]\mathtt{and}(p(\mathsf{a}), \mathtt{forall}([\mathsf{b}]q(\mathsf{b})))) \end{aligned}$$

Rewriting by NRSs

$$\begin{cases} \mathsf{a}\#Q \vdash \mathtt{and}(\mathtt{forall}([\mathsf{a}]P), Q) \to \mathtt{forall}([\mathsf{a}]\mathtt{and}(P, Q)) & (R_1) \\ \mathsf{a}\#P \vdash \mathtt{and}(P, \mathtt{forall}([\mathsf{a}]Q)) \to \mathtt{forall}([\mathsf{a}]\mathtt{and}(P, Q)) & (R_2) \end{cases}$$

$$\mathsf{b}\#P \vdash \mathtt{and}(P, \mathtt{forall}([\mathsf{b}]Q)) \to \mathtt{forall}([\mathsf{b}]\mathtt{and}(P, Q)) \quad (R_2^{(\mathsf{a}\ \mathsf{b})})$$

Permutatively renamed variant of R_2

$$\begin{aligned} & \mathtt{and}(\mathtt{forall}([\mathsf{a}]p(\mathsf{a})), \mathtt{forall}([\mathsf{a}]q(\mathsf{a}))) \\ \to_{R_1} \quad & \mathtt{forall}([\mathsf{a}]\mathtt{and}(p(\mathsf{a}), \mathtt{forall}([\mathsf{a}]q(\mathsf{a})))) \\ \approx_{\alpha} \quad & \mathtt{forall}([\mathsf{a}]\mathtt{and}(p(\mathsf{a}), \mathtt{forall}([\mathsf{b}]q(\mathsf{b})))) \\ \to_{R_2^{(\mathsf{a}\ \mathsf{b})}} \quad & \mathtt{forall}([\mathsf{a}]\mathtt{forall}([\mathsf{b}]\mathtt{and}(p(\mathsf{a}), q(\mathsf{b})))) \end{aligned}$$

Rewriting by NRSs

$$\begin{cases} \mathtt{a}\#Q \vdash \mathtt{and}(\mathtt{forall}([\mathtt{a}]P), Q) \to \mathtt{forall}([\mathtt{a}]\mathtt{and}(P,Q)) & (R_1) \\ \mathtt{a}\#P \vdash \mathtt{and}(P, \mathtt{forall}([\mathtt{a}]Q)) \to \mathtt{forall}([\mathtt{a}]\mathtt{and}(P,Q)) & (R_2) \end{cases}$$

$$\mathtt{b}\#P \vdash \mathtt{and}(P, \mathtt{forall}([\mathtt{b}]Q)) \to \mathtt{forall}([\mathtt{b}]\mathtt{and}(P,Q)) \quad (R_2^{(\mathtt{a}\ \mathtt{b})})$$

Permutatively renamed variant of R_2

$$\begin{aligned} & \mathtt{and}(\mathtt{forall}([\mathtt{a}]p(\mathtt{a})), \mathtt{forall}([\mathtt{a}]q(\mathtt{a}))) \\ \to_{R_1} \quad & \mathtt{forall}([\mathtt{a}]\mathtt{and}(p(\mathtt{a}), \mathtt{forall}([\mathtt{a}]q(\mathtt{a})))) \\ \approx_\alpha \quad & \mathtt{forall}([\mathtt{a}]\mathtt{and}(p(\mathtt{a}), \mathtt{forall}([\mathtt{b}]q(\mathtt{b})))) \\ \to_{R_2^{(\mathtt{a}\ \mathtt{b})}} \quad & \mathtt{forall}([\mathtt{a}]\mathtt{forall}([\mathtt{b}]\mathtt{and}(p(\mathtt{a}), q(\mathtt{b})))) \end{aligned}$$

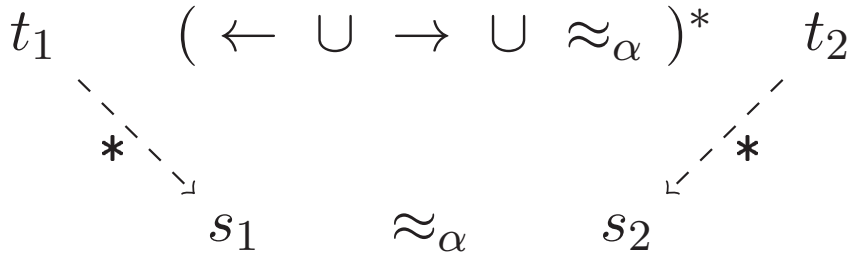
Definition [rewrite relation]

$$\Delta \vdash s \to_{\langle R,\pi,p,\sigma\rangle} t \iff^{\text{def}} \Delta \vdash \nabla^\pi\sigma,\ \Delta \vdash s|_p \approx_\alpha l^\pi\sigma,\ t = s[r^\pi\sigma]_p.$$

$$\Delta \vdash s \to_{\mathcal{R}} t \iff^{\text{def}} \Delta \vdash s \to_{\langle R,\pi,p,\sigma\rangle} t \text{ for some } R \in \mathcal{R}, \pi, p, \sigma.$$

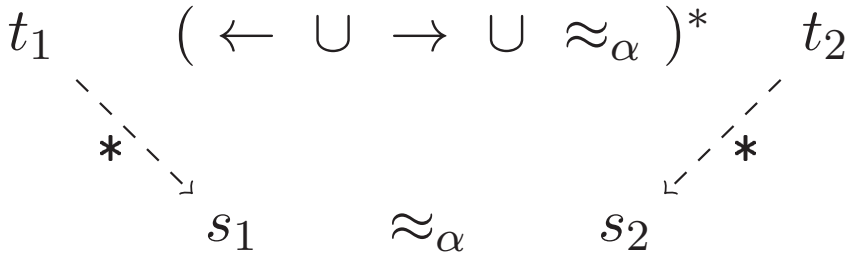
Confluence of Nominal Rewriting

Church-Rosser modulo $\approx_\alpha$ (CRM for short)



Confluence of Nominal Rewriting

Church-Rosser modulo $\approx_\alpha$ (CRM for short)

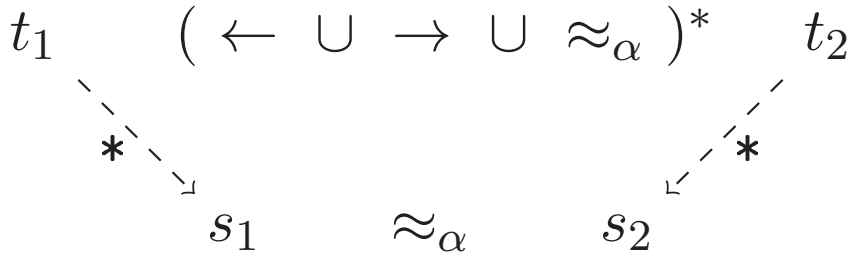


Definition

Let $\nabla_i \vdash l_i \to r_i$ $(i = 1, 2)$ be nominal rewrite rules. Suppose $\nabla_1 \cup \nabla_2^\pi \cup \{l_1 \approx l_2^\pi|_p\}$ is unifiable for some permutation π and a non-variable position p of l_2, and let $\langle \Gamma, \theta \rangle$ be an mgu. Then, $\Gamma \vdash \langle l_2^\pi\theta[r_1\theta]_p, r_2^\pi\theta \rangle$ is called a *basic critical pair* (*BCP*).

Confluence of Nominal Rewriting

Church-Rosser modulo $\approx_\alpha$ (CRM for short)

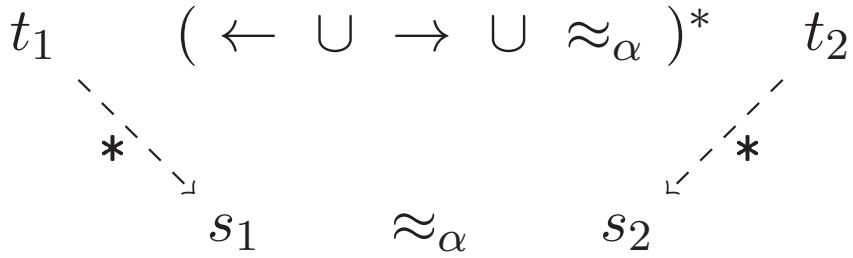


Definition

Let $\nabla_i \vdash l_i \to r_i$ $(i = 1, 2)$ be nominal rewrite rules. Suppose $\nabla_1 \cup \nabla_2^\pi \cup \{l_1 \approx l_2^\pi|_p\}$ is unifiable for some permutation π and a non-variable position p of l_2, and let $\langle \Gamma, \theta \rangle$ be an mgu. Then, $\Gamma \vdash \langle l_2^\pi\theta[r_1\theta]_p, r_2^\pi\theta \rangle$ is called a *basic critical pair* (*BCP*).

Confluence of Nominal Rewriting

Church-Rosser modulo $\approx_\alpha$ (CRM for short)



Definition

Let $\nabla_i \vdash l_i \to r_i$ $(i = 1, 2)$ be nominal rewrite rules. Suppose $\nabla_1 \cup \nabla_2^\pi \cup \{l_1 \approx l_2^\pi|_p\}$ is unifiable for some permutation π and a non-variable position p of l_2, and let $\langle \Gamma, \theta \rangle$ be an mgu. Then, $\Gamma \vdash \langle l_2^\pi\theta[r_1\theta]_p, r_2^\pi\theta \rangle$ is called a *basic critical pair* (*BCP*).

BCP is computable, by *Equivariant Nominal Unification* (Cheney, 2010), but a considerable bridge was needed (next slide).

Computation of Basic Critical Pairs

Solution of Equivariant Unification Problem:

$$\begin{cases} \{Y \mapsto (\texttt{forall}\ [\mathsf{b}]X), P : \mathsf{a} \mapsto A, P : \mathsf{b} \mapsto \mathsf{a}\}, \\ \{Y \mapsto (\texttt{forall}\ [\mathsf{a}][(\mathsf{a}\ \mathsf{b})]X), P : \mathsf{a} \mapsto A, P : \mathsf{b} \mapsto \mathsf{b}\}, \\ \{Y \mapsto (\texttt{forall}\ [(\mathsf{a}\ C)\cdot\mathsf{b}][(\mathsf{a}\ C)]X), C\#X, C \not\approx \mathsf{a}, C \not\approx \mathsf{b}, P : \mathsf{a} \mapsto A, P : \mathsf{b} \mapsto \end{cases}$$

Obtained BCP:

$$\begin{cases} \vdash \langle \texttt{forall}[\mathsf{b}]\texttt{forall}[\mathsf{b}]\texttt{forall}[\mathsf{a}]X,\ \texttt{forall}[\mathsf{a}]\texttt{forall}[\mathsf{b}]\texttt{forall}[\mathsf{b}]X \rangle \\ \vdash \langle \texttt{forall}[\mathsf{c}]\texttt{forall}[\mathsf{b}]\texttt{forall}[\mathsf{a}]X,\ \texttt{forall}[\mathsf{a}]\texttt{forall}[\mathsf{c}]\texttt{forall}[\mathsf{b}]X \rangle \\ \vdash \langle \texttt{forall}[\mathsf{a}]\texttt{forall}[\mathsf{b}]\texttt{forall}[\mathsf{a}]X,\ \texttt{forall}[\mathsf{b}]\texttt{forall}[\mathsf{a}]\texttt{forall}[\mathsf{a}][(\mathsf{a}\ \mathsf{b})]X \rangle \\ \vdash \langle \texttt{forall}[\mathsf{c}]\texttt{forall}[\mathsf{b}]\texttt{forall}[\mathsf{a}]X,\ \texttt{forall}[\mathsf{b}]\texttt{forall}[\mathsf{c}]\texttt{forall}[\mathsf{a}][(\mathsf{a}\ \mathsf{b})]X \rangle \\ \mathsf{c}\#X \vdash \langle \texttt{forall}[\mathsf{b}]\texttt{forall}[\mathsf{b}]\texttt{forall}[\mathsf{a}]X,\ \texttt{forall}[\mathsf{c}]\texttt{forall}[\mathsf{b}]\texttt{forall}[\mathsf{b}](\mathsf{a}\ \mathsf{c} \\ \mathsf{c}\#X \vdash \langle \texttt{forall}[\mathsf{a}]\texttt{forall}[\mathsf{b}]\texttt{forall}[\mathsf{a}]X,\ \texttt{forall}[\mathsf{c}]\texttt{forall}[\mathsf{a}]\texttt{forall}[\mathsf{b}](\mathsf{a}\ \mathsf{c} \\ \mathsf{d}\#X \vdash \langle \texttt{forall}[\mathsf{c}]\texttt{forall}[\mathsf{b}]\texttt{forall}[\mathsf{a}]X,\ \texttt{forall}[\mathsf{d}]\texttt{forall}[\mathsf{c}]\texttt{forall}[\mathsf{b}](\mathsf{a}\ \mathsf{d} \end{cases}$$

computation of BCP = computation of EUP + Instantiation

(But, such an instantiation is not generally possible for arbitrary EUPs.)

Implemented Confluence Criteria

Proposition [SKAT, RTA 2015]

Abstract skeleton preserving orthogonal NRSs are CRM.

Proposition [SKAT, SCSS 2016]

Linear uniform NRSs are CRM if $\Gamma \vdash u \to^{=} \circ \approx_{\alpha} \circ \leftarrow^{*} v$ and $\Gamma \vdash u \to^{*} \circ \approx_{\alpha} \circ \leftarrow^{=} v$ for any BCP $\Gamma \vdash \langle u, v \rangle$.

Proposition [SKAT, SCSS 2016]

Terminating uniform NRSs are CRM if and only if $\Gamma \vdash u \, (\to^{*} \circ \approx_{\alpha} \circ \leftarrow^{*}) \, v$ for any BCP $\Gamma \vdash \langle u, v \rangle$.

Proposition [KAT, 2016]

Left-linear uniform NRSs are CRM if $\Gamma \vdash u \twoheadrightarrow \circ \approx_{\alpha} v$ $(u \twoheadrightarrow \circ \approx_{\alpha} \circ \leftarrow^{*} v)$ for any inner (resp. outer) BCP $\Gamma \vdash \langle u, v \rangle$.

Implemented Confluence Criteria

Proposition [SKAT, RTA 2015]

Abstract skeleton preserving orthogonal NRSs are CRM.

Proposition [SKAT, SCSS 2016]

Linear uniform NRSs are CRM if $\Gamma \vdash u \to^{=} \circ \approx_{\alpha} \circ \leftarrow^{*} v$ and $\Gamma \vdash u \to^{*} \circ \approx_{\alpha} \circ \leftarrow^{=} v$ for any BCP $\Gamma \vdash \langle u, v \rangle$.

Proposition [SKAT, SCSS 2016]

Terminating uniform NRSs are CRM if and only if $\Gamma \vdash u \; (\to^{*} \circ \approx_{\alpha} \circ \leftarrow^{*}) \; v$ for any BCP $\Gamma \vdash \langle u, v \rangle$.

Termination proving via encoding to f.o.rewriting is given.

Proposition [KAT, 2016]

Left-linear uniform NRSs are CRM if $\Gamma \vdash u \twoheadrightarrow \circ \approx_{\alpha} v$ $(u \twoheadrightarrow \circ \approx_{\alpha} \circ \leftarrow^{*} v)$ for any inner (resp. outer) BCP $\Gamma \vdash \langle u, v \rangle$.

nrbox: **n**ominal **r**ewriting tool**box**

- Written in SML/NJ, about 4500 loc
- http://www.nue.ie.niigata-u.ac.jp/tools/nrbox/
- Requires an external *termination prover* for first-order TRSs.
- Some approximations of $\overset{*}{\rightarrow}$ is employed for checking criteria.
- Tested on a collection of 30 NRSs (successful in 20 NRSs).

```
$ sml @SMLload=nrbox.x86-linux  examples/1.nrs
YES
NRS:
   [ (0): b#X |- (lam [a][]X) -> (lam [b][(a b)]X) ]
Check confluence by orthogonality
not abstract skeleton preserving
Check confluence by strongly closed criterion
linear
uniform
all BCPs are strongly closed
result: YES
time: 3 msec.
$
```

Experiments

	NRS	Orth.	Strong	K.-B.	Parallel
1	α-reduction rule	MAYBE	YES	MAYBE	YES
2	Eta: η-reduction rule	YES	YES	YES	YES
3	η-expansion rule	MAYBE	MAYBE	MAYBE	MAYBE
4	$\mathcal{R}^*_\sigma$: subst. for λ with σ_ε	MAYBE	MAYBE	YES	YES
5	β-reduction $\{\text{Beta}\} \cup \mathcal{R}^*_\sigma$	MAYBE	MAYBE	MAYBE	MAYBE
6	a fragment of ML	MAYBE	MAYBE	MAYBE	MAYBE
7	PNF of FOF	MAYBE	MAYBE	NO	MAYBE
8	PNF of FOF with addition	MAYBE	MAYBE	NO	MAYBE
9	non-joinable trivial CP	MAYBE	MAYBE	MAYBE	MAYBE
10	$\{a\#X \vdash X \to [a]X\}$	MAYBE	MAYBE	MAYBE	MAYBE
11	$\{\text{Eta}, \bot\}$	MAYBE	MAYBE	NO	MAYBE
12	$\{\text{Eta}, \bot\}$ with CP	MAYBE	YES	YES	YES
13	summation	MAYBE	MAYBE	NO	MAYBE
14	summation with CP	MAYBE	MAYBE	YES	MAYBE
15	$\{\vdash f(X) \to [a]X\}$	MAYBE	MAYBE	NO	MAYBE
16	$\{a\#X \vdash f(X) \to [a]X\}$	YES	YES	YES	YES
17	$\mathcal{R}_\sigma$: subst. for λ with $\sigma_{\text{var}\varepsilon}$	YES	MAYBE	YES	YES
18	β-reduction $\{\text{Beta}\} \cup \mathcal{R}_\sigma$	MAYBE	MAYBE	MAYBE	MAYBE
19	$\beta\eta$-reduction $\{\text{Beta}\} \cup \{\text{Eta}\} \cup \mathcal{R}_\sigma$	MAYBE	MAYBE	MAYBE	MAYBE
20	$\mathcal{R}_{\text{uc-}\eta}$	MAYBE	MAYBE	MAYBE	MAYBE
21	$\mathcal{R}_{\text{uc-}\eta\text{-exp}}$	MAYBE	MAYBE	NO	MAYBE
22	μ-substitution for $\lambda\mu$-term	YES	MAYBE	YES	YES
23	$\{\vdash f(X) \to f([a]X)\}$	MAYBE	MAYBE	MAYBE	MAYBE
24	NNF of $\{\neg, \forall, \wedge\}$-form. with swap	MAYBE	YES	YES	YES
25	$\text{Com}_\forall$: com. rule for $\forall$	MAYBE	YES	MAYBE	MAYBE
26	PNF of $\{\forall, \wedge\}$-form.	MAYBE	MAYBE	NO	MAYBE
27	PNF of $\{\forall, \wedge\}$-form. + $\text{Com}_\forall$	MAYBE	MAYBE	MAYBE	MAYBE
28	NNF of $\{\neg, \forall, \exists\}$-form.	MAYBE	MAYBE	YES	MAYBE
29	NNF of FOF	MAYBE	MAYBE	YES	MAYBE
30	NNF of FOF without DNE	YES	YES	YES	YES
	(♯YES, ♯NO)	(5,0)	(7,0)	(11,7)	(9,0)
	$\sum$ time (msec.)	611	1367	4377	2217

Conclusion

- "First" confluence tool for nominal rewriting has been developed (ex. our successor tool).
- Solved a gap to move from equivariant unification to critical pair computation.
  - (Canonical set of) critical pairs is obtained by instantiation of the solution of equivariant unification.
- We propose a termination proving techique for NRSs based on encoding (to that of first-order TRSs).
- 30 NRSs are collected from literature for testing and the prover succeeds in 20 NRSs (13 for confluence, 7 for non-confluence).
- Helpful to reasoning about nominal rewriting theory.

Future plan

- Revisiting the notion of nominal rewriting; how one should deal with freshness constraint? ($\Rightarrow$ our SCSS 2016 paper)
- Extension to termination tool and completion tool.