Logical Foundations of Multiset Rewriting

Iliano Cervesato iliano@itd.nrl.navy.mil

ITT Industries, inc @ NRL Washington, DC

http://www.cs.stanford.edu/~iliano

Outline

- Motivations
- Propositional multiset rewriting
  - Interpretation in linear logic
  - Interpretation as linear logic
- Logical extension
  - First-order multiset rewriting
  - ω-multisets
- Applications
  - Specification of security protocols
  - A bridge to process algebra

Motivations

- Multiset rewriting (a.k.a. Petri nets)
  - Fundamental model of distributed computing
    - Competitor: Process Algebras
  - Basis for security protocol spec. languages
    - MSR family
    - … several others
  - Many extensions, more or less ad hoc
- Shallow relations to logic
  - Simple encodings
  - No deep insight

This Work

- Show that multiset rewriting has deeper relations to logic
  - Interpretation <u>as</u> logic, rather than
  - Interpretation <u>in</u> logic
- Explain and rationalize extensions
- Better specification languages
- Bridge to process algebra

Multiset Rewriting

- Multiset: set with repetitions allowed

$$\underline{a} \;::=\; \bullet \;\mid\; a,\, \underline{a}$$

  - ➢ Commutative monoid
    - "," is operation
    - "•" is identity

      ("," is commutative, associative, with "•" as unit)

- Rewrite rule:

$$\underline{a} \rightarrow \underline{b}$$

  - ➢ Monoidal rewriting

Semantics of Multiset Rewriting

- Base step: $\underline{s} \rightarrow_R \underline{s}'$

$$\boxed{\underline{a}, \underline{s} \rightarrow_{R,\, (\underline{a} \rightarrow \underline{b})} \underline{b}, \underline{s}}$$

- Reachability

  $\underline{s}_{\underline{0}} \rightarrow^*_R \underline{s}_{\underline{n}}$

  - Iteration of $\rightarrow$
  - R&T closure of $\rightarrow$

- Infinity

  $\underline{s}_{\underline{0}} \rightarrow^*_R$

  - Limit of $\_ \rightarrow^*_R \_$

Linear Logic

Logic with formulas as resources

- Formulas

$$A \ ::= \ a \ | \ A \otimes A \ | \ 1 \ | \ A \multimap A \ | \ \ldots$$

- Judgment (DILL / LV sequent)

$$\Gamma; \Delta \dashrightarrow A$$

Unrestricted context

- subject to exchange, weakening and contraction
- behaves like context in traditional logic

Linear context

- subject to exchange only

Some Rules

$$\frac{\Gamma;\ \Delta,\ A,\ B \dashrightarrow C}{\Gamma;\ \Delta,\ A \otimes B \dashrightarrow C} \qquad \frac{\Gamma;\ \Delta_1 \dashrightarrow A \quad \Gamma;\ \Delta_2 \dashrightarrow B}{\Gamma;\ \Delta_1,\ \Delta_2 \dashrightarrow A \otimes B}$$

$$\frac{\Gamma;\ \Delta \dashrightarrow C}{\Gamma;\ \Delta,\ 1 \dashrightarrow C} \qquad \frac{}{\Gamma;\ \cdot \dashrightarrow 1}$$

$$\frac{\Gamma;\ \Delta_1 \dashrightarrow A \quad \Gamma;\ \Delta_2,\ B \dashrightarrow C}{\Gamma;\ \Delta_1,\ \Delta_2\ ,\ A \multimap B \dashrightarrow C} \qquad \frac{\Gamma;\ \Delta,\ A \dashrightarrow B}{\Gamma;\ \Delta \dashrightarrow A \multimap B}$$

$$\frac{\Gamma,\ A;\ \Delta,\ A \dashrightarrow C}{\Gamma,\ A;\ \Delta \dashrightarrow C} \qquad \frac{}{\Gamma;\ A \dashrightarrow A}$$

LL Interpretation of MSR

- Several possibilities
  - "Conjunctive" encoding
- Objective

$$
\begin{tikzcd}
R \arrow[d, dashed] & ; & \underline{s}_0 \arrow[d, dashed] & \to^{*} \arrow[d, dashed] & \underline{s}_n \arrow[d, dashed] \\
\Gamma & ; & \Delta & \dashrightarrow & A
\end{tikzcd}
$$

  - Reachability mapped to derivability

Encoding

- R
  - $\rightarrow \;\Rightarrow\; \multimap$
- s_n
  - $, \;\Rightarrow\; \otimes$
  - $\bullet \;\Rightarrow\; 1$
- s_0
  - $, \;\Rightarrow\; ,$
  - $\bullet \;\Rightarrow\; \bullet$

  … or like s_n

Encoding

- R
  - $[a \rightarrow b] = [a] \multimap [b]$

- s_n
  - $[a] = a$
  - $[\bullet] = 1$
  - $[\underline{a}\,,\,\underline{b}] = [\underline{a}] \otimes [\underline{b}]$

- s_0
  - $[[a]] = a$
  - $[[\bullet]] = \bullet$ or $[\bullet]$
  - $[[\underline{a}\,,\,\underline{b}]] = [[a]]\,,\,[[b]]$ or $[\underline{a}\,,\,\underline{b}]$

  Well defined because
  - $(\Delta s, \text{“,”}, \text{“}\bullet\text{”})$ is a commutative monoid
  - $(As, \otimes, 1)_{/\,\text{-->}}$ is a commutative monoid

Property

$s_0 \rightarrow^*_R s_n$ iff $[R] ; [[s_0]] \dashrightarrow [s_n]$

- For appropriate inverse encodings

$\Gamma; A \dashrightarrow B$ iff $[A] \rightarrow^*_{[\Gamma]} [B]$

Encoding of MSR in LL

End of the Story ?

- ~~Yes~~ NO!
- From interpretation of MSR <u>in</u> logic to interpretation of MSR <u>as</u> logic
- Multiset rewriting semantics = left sequent rules
- First, a few rough edges to smooth

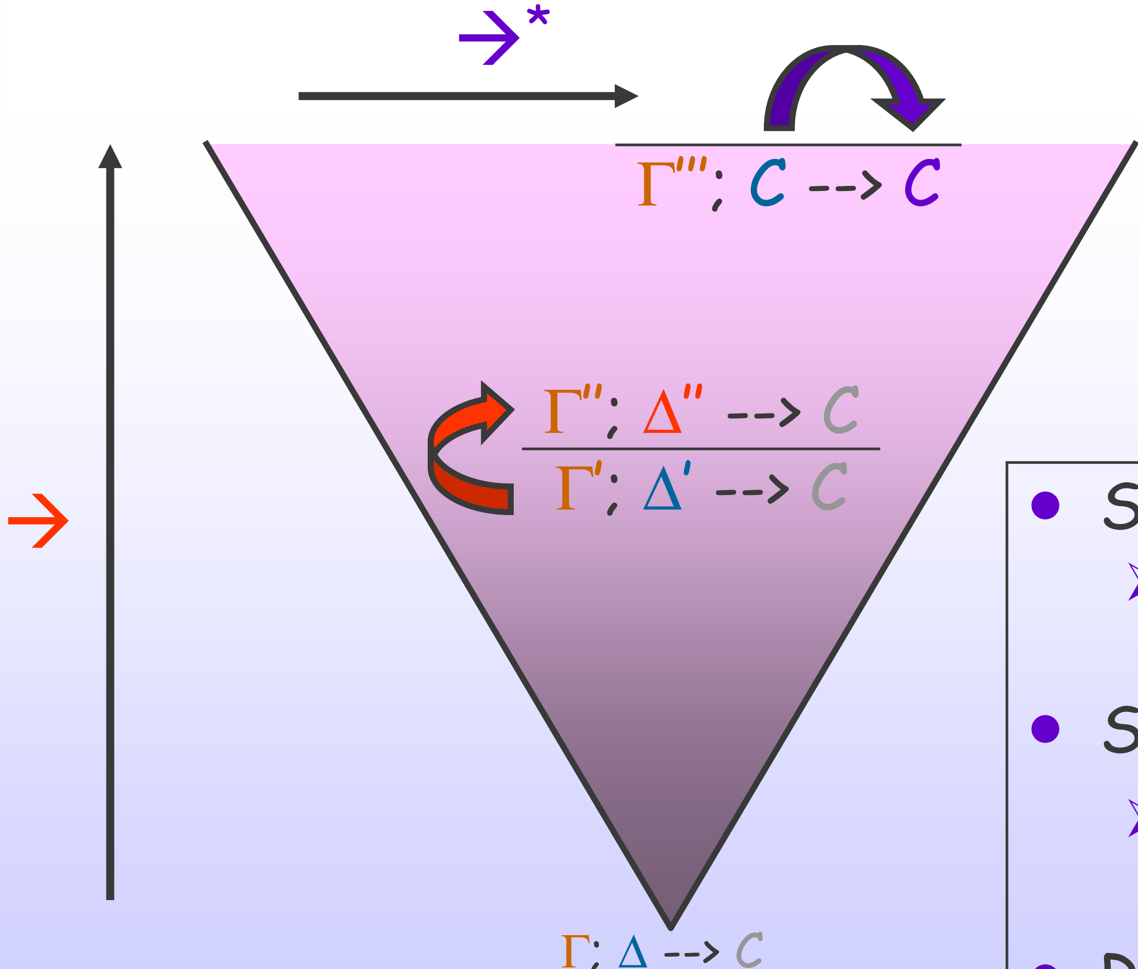
Context vs. Formulas (1)

- Either go against tradition of logic
  - (As, $\otimes$, 1) is a congruence w.r.t. derivability
- ➢ Identify contexts and formulas
  - Whenever formula is expected
    - Turn , into $\otimes$
    - Turn • into 1
  - Consistent with categorical semantics of logic
  - Has to be done with extreme care

Context vs. Formulas (2)

- … or go against tradition of rewriting
  - Distinguish states and multisets
    - state constructors: , and •
    - mset constructors: $\otimes$ and 1
  - Additional transition rules
    - $\underline{s}, a \otimes b \rightarrow_R \underline{s}, a, b$
    - $\underline{s}, 1 \rightarrow_R \underline{s}$
- This research is compatible with both
  - We will lean towards (2)

Rewriting View of Derivations



- Step up: →
  - Left rules
- Step across: →*
  - Axiom
- Right rules not used

Rewriting Semantics as Left Rules

$\underline{s} \rightarrow^{*}_{R} \underline{s}$	$\dfrac{}{\Gamma; A \dashrightarrow A}$
$\underline{s}, a \otimes b \rightarrow_{R} \underline{s}, a, b$	$\dfrac{\Gamma; \Delta, A, B \dashrightarrow C}{\Gamma; \Delta, A \otimes B \dashrightarrow C}$
$\underline{s}, 1 \rightarrow_{R} \underline{s}$	$\dfrac{\Gamma; \Delta \dashrightarrow C}{\Gamma; \Delta, 1 \dashrightarrow C}$
$\underline{s}, a \rightarrow_{R, (a \rightarrow b)} \underline{s}, b$	$\dfrac{\Gamma, A \multimap B; \Delta, B \dashrightarrow C}{\Gamma, A \multimap B; A, \Delta \dashrightarrow C}$

- Not quite, but not too far off
  - Admissible rule

$$\dfrac{\Gamma, A; \Delta, A \dashrightarrow C}{\Gamma, A; \Delta \dashrightarrow C} \quad + \quad \dfrac{\Gamma; \Delta_1 \dashrightarrow A \qquad \Gamma; \Delta_2, B \dashrightarrow C}{\Gamma; \Delta_1, \Delta_2, A \multimap B \dashrightarrow C}$$

Questions

- Can we make the correspondence precise?
  - Yes
- Does it extend to other connectives?
  - Yes … to a large extent
- What are the implications?
  - Logical explanation of multiset rewriting
    - Not just interpretation
    - Now MSR is logic
  - Guideline to design rewrite systems
    - Can we do this with other logics?
  - Derivations do not need to be finite
    - Goal is important only for reachability

First Proof of Concept

- First-Order Multiset Rewriting (MSR 1.0)
  - Multiset elements are FO atomic formulas
  - Rules have the form

$$\forall x_1 \ldots x_n.\ \underline{a}(\mathbf{x}) \rightarrow \exists y_1 \ldots y_k.\ \underline{b}(\mathbf{x},\mathbf{y})$$

  - Semantics $(\rightarrow^*)$

$$\boxed{\Sigma\ ;\ \underline{a}(\mathbf{t}),\ \underline{s}\ \ \rightarrow_{R,\ (\underline{a}(\mathbf{x}) \rightarrow \exists \mathbf{y}.\ \underline{b}(\mathbf{x},\mathbf{y}))}\ \ \Sigma,\mathbf{y}\ ;\ \underline{b}(\mathbf{t},\mathbf{y}),\ \underline{s} \quad \text{if } \Sigma \mid\!-\ \mathbf{t}}$$

  - Encoding is simple extension of prop. case

Semantics from Left Rules

- Updated judgment forms
  - $\Sigma ; \underline{s} \rightarrow_R \Sigma ; \underline{s}$
  - $\Gamma; \Delta \dashrightarrow_\Sigma C$
- Semantics ($\rightarrow$**)

…	…
$\Sigma ; \underline{s}, \forall x.a \rightarrow_R \Sigma ; \underline{s}, [t/x]a$ if $\Sigma \vdash t$	$\dfrac{\Gamma; \Delta, [t/x]A \dashrightarrow_\Sigma C \quad \Sigma \vdash t}{\Gamma; \Delta, \forall x.A \dashrightarrow_\Sigma C}$
$\Sigma ; \underline{s}, \exists x.a \rightarrow_R \Sigma, x ; \underline{s}, a$	$\dfrac{\Gamma; \Delta, A \dashrightarrow_{\Sigma,x} C}{\Gamma; \Delta, \exists x.A \dashrightarrow_\Sigma C}$

Comparing Semantics

Lemma

- If $\underline{a} \rightarrow^{*}_{R} (\underline{b})$, then $\underline{a} \rightarrow^{**}_{R} (\underline{b})$
- And viceversa
  - Careful with non-observable steps

Second Proof of Concept

- Minimal ω-multiset rewriting
  - Language

    $$\omega \ ::= \ a \ | \ \bullet \ | \ \omega, \omega \ | \ \omega \rightarrow \omega$$

    - No distinction between atoms and formulas
  - Semantics (v.1)
    - $\underline{s}, (a \rightarrow b), a \ \rightarrow \ \underline{s}, b$
  - Check against left rule for $\multimap$

    $$\frac{\Delta_1 \dashrightarrow A \qquad \Delta_2, B \dashrightarrow C}{\Delta_1, \Delta_2, A \multimap B \dashrightarrow C}$$

  - Semantics (v.2)
    - $\underline{s}_1, \underline{s}_2, (a \rightarrow b) \ \rightarrow \ \underline{s}_2, b \qquad \text{if} \quad \underline{s}_1 \ \rightarrow^* \ a$
    - Step depends on reachability!

Comparing Semantics

- Lemma

  $\underline{a} \rightarrow^*_{v.1} (\underline{b})$ iff $\underline{a} \rightarrow^*_{v.2} (\underline{b})$

  (⇒) Trivial by reflexivity

  (⇐) Recursively turn every step

  - $\underline{s}_1, \underline{s}_2, (a \to b) \rightarrow_{v.2} \underline{s}_2, b$ if $\underline{s}_1 \rightarrow^*_{v.2} a$

  into

  - $\underline{s}_1, \underline{s}_2, (a \to b) \rightarrow^*_{v.1} a, \underline{s}_2, (a \to b) \rightarrow_{v.1} \underline{s}_2, b$

- However
  - Do all extensions support transformation?
    - Use v.1 when adequate, v.2 other times
  - Seems to be an instance of cut elimination
    - (see later)

Adding Persistent Multisets

- Language

  $\omega \ ::= \ a \mid \bullet \mid \omega, \omega \mid \omega \rightarrow \omega \mid \forall x.\ \omega \mid \exists x.\ \omega \mid \ !\,\omega$

- Judgment
  - $\Sigma \,;\, p \,;\, \underline{s} \rightarrow \Sigma \,;\, p \,;\, \underline{s}$

- Semantics from left rules

…	…
$\Sigma \,;\, p \,;\, \underline{s}, !a \rightarrow \Sigma \,;\, p, a \,;\, \underline{s}$	$\dfrac{\Gamma, A; \Delta \dashrightarrow_\Sigma C}{\Gamma; \Delta, !A \dashrightarrow_\Sigma C}$
$\Sigma \,;\, p, a \,;\, \underline{s} \rightarrow \Sigma \,;\, p, a \,;\, \underline{s}, a$	$\dfrac{\Gamma, A; \Delta, A \dashrightarrow_\Sigma C}{\Gamma, A; \Delta \dashrightarrow_\Sigma C}$

A Word of Caution

$$!(a \otimes b) \neq \; !a \otimes !b$$

a and b can be used only together

a and b can be used independently

- $\otimes$ corresponds to “,” in Δ, but not in Γ
  - ➢ Distinguish $\otimes$ and “,” in ωMSR
  - ➢ Consider only sublanguages
  - ➢ Use different symbol “,,” in p
    - p is multiset of multisets, not multiset

Additive Conjunction and Unit

- Language

  ω ::= … | $\omega \,\&\, \omega$ | $\top$

- Semantics from left rules

…	…
$\Sigma ; p ; \underline{s}, a_1 \,\&\, a_2 \rightarrow \Sigma ; p ; \underline{s}, a_i$ Non-deterministic choice • Usually written +	$\dfrac{\Gamma; \Delta , A_i \dashrightarrow_\Sigma C}{\Gamma; \Delta , A_1 \,\&\, A_2 \dashrightarrow_\Sigma C}$
(no $\top$-transition) Absence of any choice	(no left rule)

Additive Disjunction and Unit

- Language

  $\omega \ ::= \ \ldots \ | \ \omega \oplus \omega \ | \ 0$

- Semantics from left rules

$\Sigma \,;\, p \,;\, \underline{s}, 0 \rightarrow^* \ \underline{s}_n$

$$\frac{}{\Gamma ;\, \Delta \,,\, 0 \dashrightarrow_{\Sigma} C}$$

- ➢Inconsistency?
- ➢Forced reachability?

The case of ⊕

$$\frac{\Gamma;\ \Delta,\ A \dashrightarrow_{\Sigma} C \qquad \Gamma;\ \Delta,\ B \dashrightarrow_{\Sigma} C}{\Gamma;\ \Delta,\ A \oplus B \dashrightarrow_{\Sigma} C}$$

$$\Sigma\ ;\ p\ ;\ \underline{s},\ a \oplus b \rightarrow \begin{cases} \Sigma\ ;\ p\ ;\ \underline{s},\ a \rightarrow^{*} (c) \\ \Sigma\ ;\ p\ ;\ \underline{s},\ b \rightarrow^{*} (c) \end{cases}$$

- The 2 computations shall be synchronized
  - If one “ends”, the other “ends” in the same way
    - Breakpoint, or final state
  - If one diverges, the other shall diverge
- Flavor of
  - Confluence
  - Bisimulation?

Multiplicative Disjunction and Unit

- Language:

  ω ::= … | $\omega \wp \omega$ | $\perp$

- Semantics from left rules

Σ ; p ; $\perp$ →* •

$$\frac{}{\Gamma ; \perp \dashrightarrow_{\Sigma} \bullet}$$

  - Abort?
  - Deadlock?

The Case of $\wp$

$$\frac{\Gamma; \Delta_1, A \dashrightarrow_{\Sigma} \Psi_1 \qquad \Gamma; \Delta_2, B \dashrightarrow_{\Sigma} \Psi_2}{\Gamma; \Delta_1, \Delta_2, A \wp B \dashrightarrow_{\Sigma} \Psi_1, \Psi_2}$$

$$\Sigma ; p ; \underline{s}_1, \underline{s}_2, a \wp b \rightarrow \begin{cases} \Sigma ; p ; \underline{s}_1, a \\ \\ \Sigma ; p ; \underline{s}_2, b \end{cases}$$

- Start of completely independent computations involving a and b

The Axiom Rule

$$\overline{\Gamma; A \dashrightarrow_{\Sigma} A}$$

$$\Sigma\ ;\ p\ ;\ a\ \rightarrow^{*}\ a$$

- Makes a reachability statement
- Turns $\rightarrow$ into $\rightarrow^{*}$ a

The Cut Rules

$$\frac{\Gamma;\ \Delta_1 \mathrel{\text{-->}}_{\Sigma} A \qquad \Gamma;\ \Delta_2, A \mathrel{\text{-->}}_{\Sigma} C}{\Gamma;\ \Delta_1, \Delta_2 \mathrel{\text{-->}}_{\Sigma} C}$$

$$\frac{\Gamma;\ \bullet \mathrel{\text{-->}}_{\Sigma} A \qquad \Gamma, A;\ \Delta \mathrel{\text{-->}}_{\Sigma} C}{\Gamma;\ \Delta \mathrel{\text{-->}}_{\Sigma} C}$$

$$\Sigma \,;\, p \,;\, \underline{s_1}, \underline{s_2} \rightarrow \Sigma \,;\, p \,;\, a, \underline{s_2} \qquad \text{if}\ \ \Sigma \,;\, p \,;\, \underline{s_2} \rightarrow^{*} a$$

$$\Sigma \,;\, p \,;\, \underline{s} \rightarrow \Sigma \,;\, p, a \,;\, \underline{s} \qquad \text{if}\ \ \Sigma \,;\, p \,;\, \bullet \rightarrow^{*} a$$

- Compositionality laws

- Does cut elimination hold?
- Note
  - Not as deep as in Logic
    - No right rules

Summary: ω-Multisets

ω	$::=$	a	atomic object
	$\mid$	$\bullet$	empty mset
	$\mid$	ω, ω	mset formation
	$\mid$	$\omega \rightarrow \omega$	mset rewrite
	$\mid$	$\top$	no-op
	$\mid$	$\omega + \omega$	choice
	$\mid$	$!\,\omega$	replication
	$\mid$	$\forall x.\ \omega$	instantiation
	$\mid$	$\exists x.\ \omega$	generation
	$\mid$	$0 \mid \omega \oplus \omega \mid \bot \mid \omega \wp \omega$	
	$\mid$	$?\,\omega \mid \omega^{\perp}$	???

Summary: ω-Multisets Semantics

•	$\Sigma ; \underline{p} ; (\underline{s}, 1)$	→	$\Sigma ; \underline{p} ; \underline{s}$
,	$\Sigma ; \underline{p} ; (\underline{s}, a \otimes b)$	→	$\Sigma ; \underline{p} ; (\underline{s}, a, b)$
$\rightarrow$	$\Sigma ; \underline{p} ; (\underline{s}, a, a \rightarrow b)$	→	$\Sigma ; \underline{p} ; (\underline{s}, b)$
$\top$		(no rule)	
&	$\Sigma ; \underline{p} ; (\underline{s}, a_1 \,\&\, a_2)$	→	$\Sigma ; \underline{p} ; (\underline{s}, a_i)$
!	$\Sigma ; \underline{p} ; (\underline{s}, !a)$	→	$\Sigma ; (\underline{p}, a) ; \underline{s}$
$\forall$	$\Sigma ; \underline{p} ; (\underline{s}, \forall x.\, a)$	→	$\Sigma ; \underline{p} ; (\underline{s}, [t/x]\underline{a})$
$\exists$	$\Sigma ; \underline{p} ; (\underline{s}, \exists x.\, a)$	→	$(\Sigma, x) ; \underline{p} ; (\underline{s}, a)$
	$\Sigma ; (\underline{p}, a) ; \underline{s}$	→	$\Sigma ; (\underline{p}, \underline{a}) ; (\underline{s}, a)$

Applications to Security

- MSR: family of security protocol specification languages
  - MSR 1: first-order multiset rewriting
  - MSR 2: MSR 1 + dependent types
  - MSR 3: ω-multiset (+ dependent types)
- Unified logical view
  - Better understanding of where we are
  - Hint about where to go next

NSPK in MSR 2.0

$A \rightarrow B: \{N_A, A\}_{KB}$
$B \rightarrow A: \{N_A, N_B\}_{KA}$
$A \rightarrow B: \{N_B\}_{KB}$

$\forall A$: princ.
{ $\exists L$: princ $\times$ $\sum B$:princ.pubK B $\times$ nonce $\rightarrow$ mset.

$\forall B$: princ. $\forall K_B$: pubK B.

•

$\rightarrow \exists N_A$: nonce.

net $(\{N_A, A\}_{KB})$, $L(A, B, K_B, N_A)$

$\forall B$: princ. $\forall K_B$: pubK B.
$\forall K_A$: pubK A. $\forall K_A{}'$: prvK K_A.
$\forall N_A$: nonce. $\forall N_B$: nonce.

net $(\{N_A, N_B\}_{KA})$, $L(A, B, K_B, N_A)$

$\rightarrow$ net $(\{N_B\}_{KB})$

}

NSPK in MSR 3

$A \rightarrow B$: $\{N_A, A\}_{KB}$
$B \rightarrow A$: $\{N_A, N_B\}_{KA}$
$A \rightarrow B$: $\{N_B\}_{KB}$

$\forall A$:princ.
$\forall B$: princ. $\forall K_B$: pubK B.

•

$\rightarrow \exists N_A$: nonce.
net $(\{N_A, A\}_{KB})$,
$(\forall K_A$: pubK A. $\forall K_A'$: prvK K_A. $\forall N_B$: nonce.
net $(\{N_A, N_B\}_{KA})$
$\rightarrow$ net $(\{N_B\}_{KB}))$

MSR 3.0

- Succinct representations
  - Simpler specifications
  - Economy of reasoning
- Logical foundations
- Bridge between
  - State-based representation
  - Process-based representations
    - Logical foundation of process algebra?

MSR vs. Process Algebra

MSR

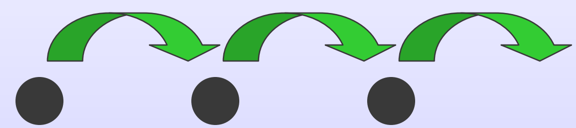
- NRL Prot. Analyzer, CAPSL/CIL, Paulson's approach, …

and Process Algebra

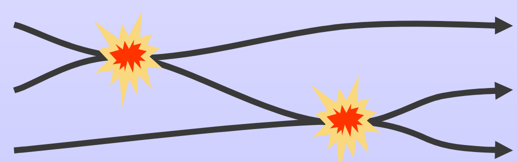
- Strand spaces, spi-calculus, other process-based lang.

operate in <u>very different</u> ways:

- State transitions

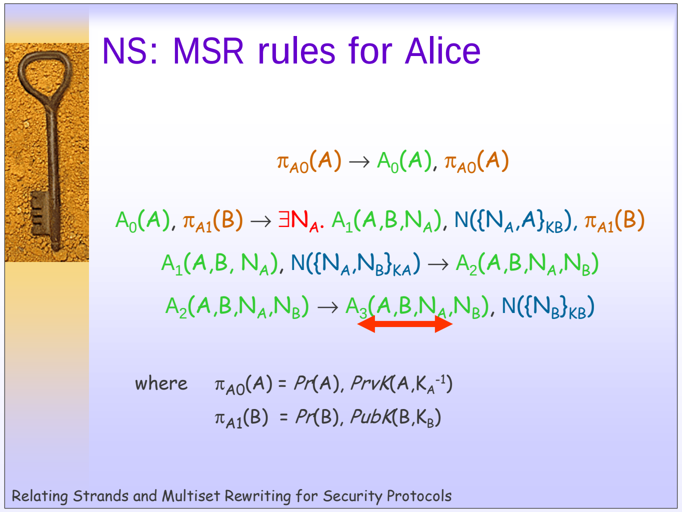


- Contact evolution

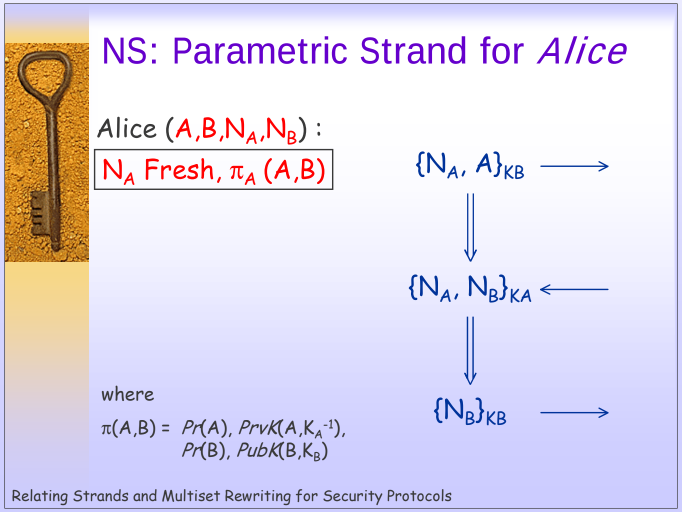


Representing Protocols

- MSR 2 $\left[\begin{array}{c} n \rightarrow a_1, n' \\ n'', a_1 \rightarrow a_2, n''' \\ \ldots \end{array}\right]$
  - a_i pass control/data to the next rule



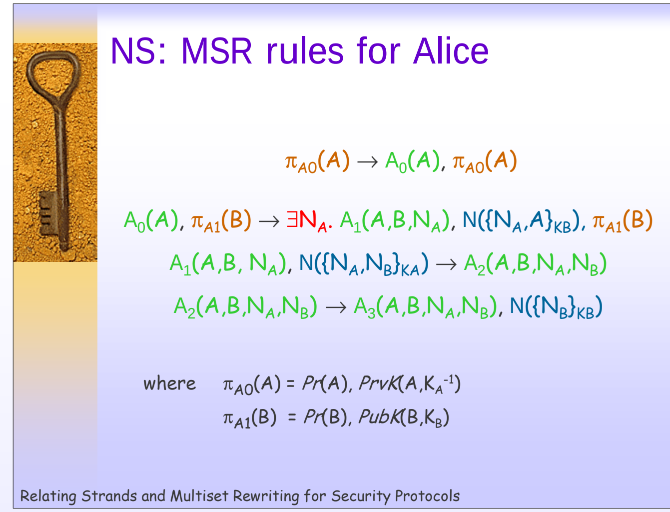
- PA $\underline{n}.n'.\underline{n''}.n'''. \ldots .0$
  - Control is implicit

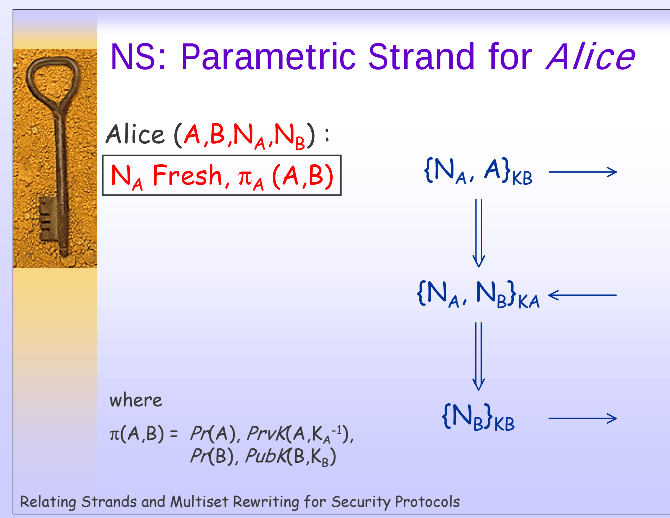


Representing Protocols

- **MSR 2** $\left[\begin{array}{c} n \rightarrow a_1, n' \\ n'', a_1 \rightarrow a_2, n''' \\ \ldots \end{array}\right]$
  - a_i pass control/data to the next rule

- **MSR 3** $n \rightarrow n', (n'' \rightarrow n''', (\ldots))$
  - Control is implicit

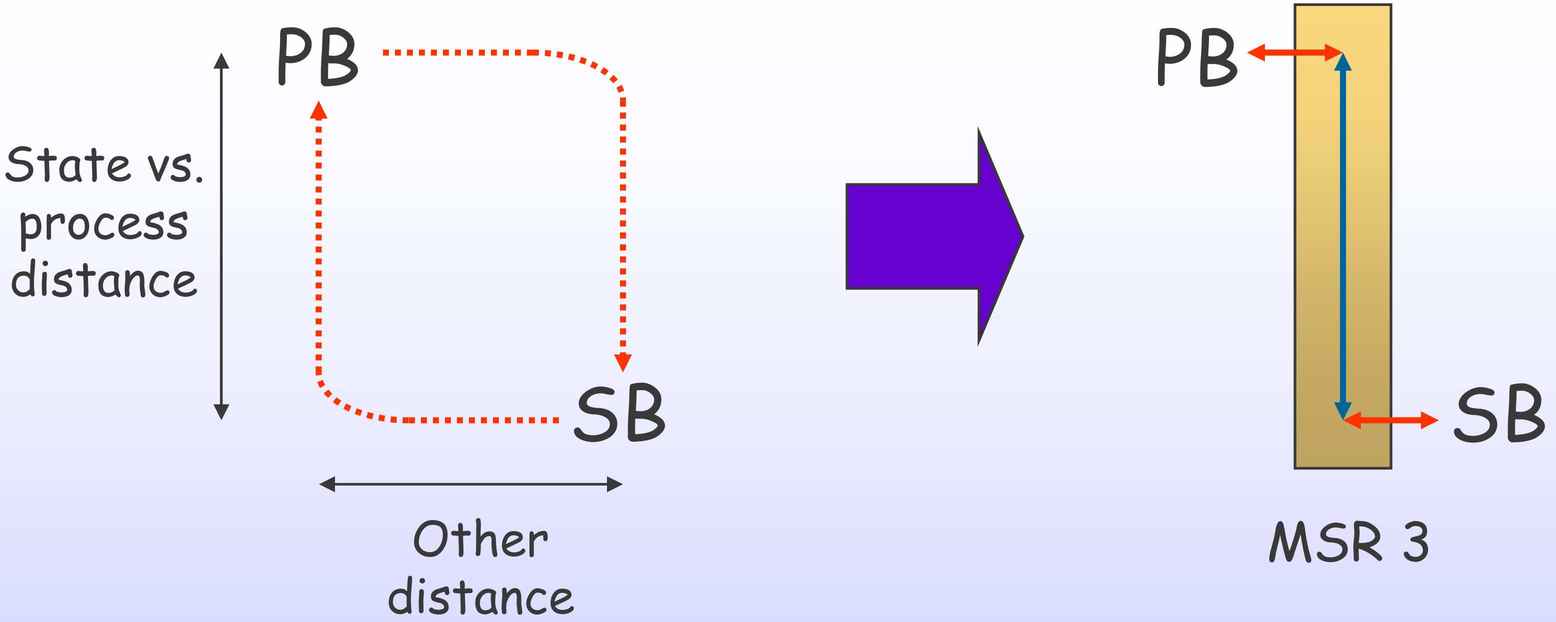




ω-Multisets and Process Algebra

- Similarities
  - ω-Multisets behave like very general process algebra
    - π-calculus
    - Join calculus
- Differences
  - PA's structural equivalences
- Towards a logical foundation of Process Algebra?

Encoding Distributed Algorithms



State ↔ Process translation done once and forall

Conclusions

- Interpretation of multiset rewriting guided by left rules of linear logic
- Definition of ω-multisets
- Hint at application in security protocol specification
  - MSR 3.0
- Possible relationship with process algebras