An Automata Based Approach for Verification of Information Flow Properties

Deepak D'Souza, Raghavendra K.R., Barbara Sprick

Indian Institute of Science, Bangalore, India

Background

Granting, restricting & controlling the flow of information

Background

Granting, restricting & controlling the flow of information

Goguen, Meseguer, ’82 - Non-Interference

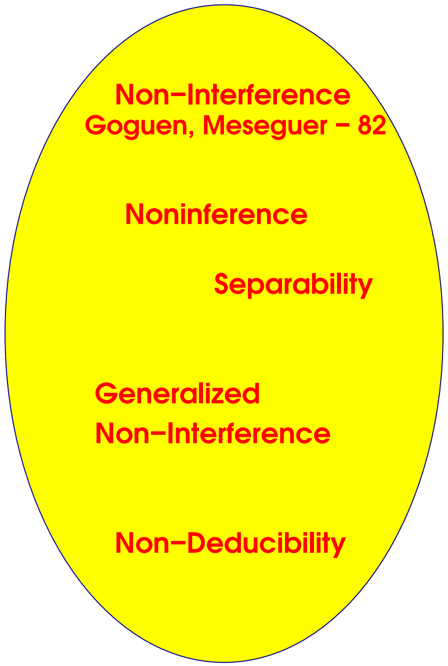
"What one group of users does has no effect on what other group of users does"

Background

Granting, restricting & controlling the flow of information

Goguen, Meseguer, ’82 - Non-Interference

"What one group of users does has no effect on what other group of users does"

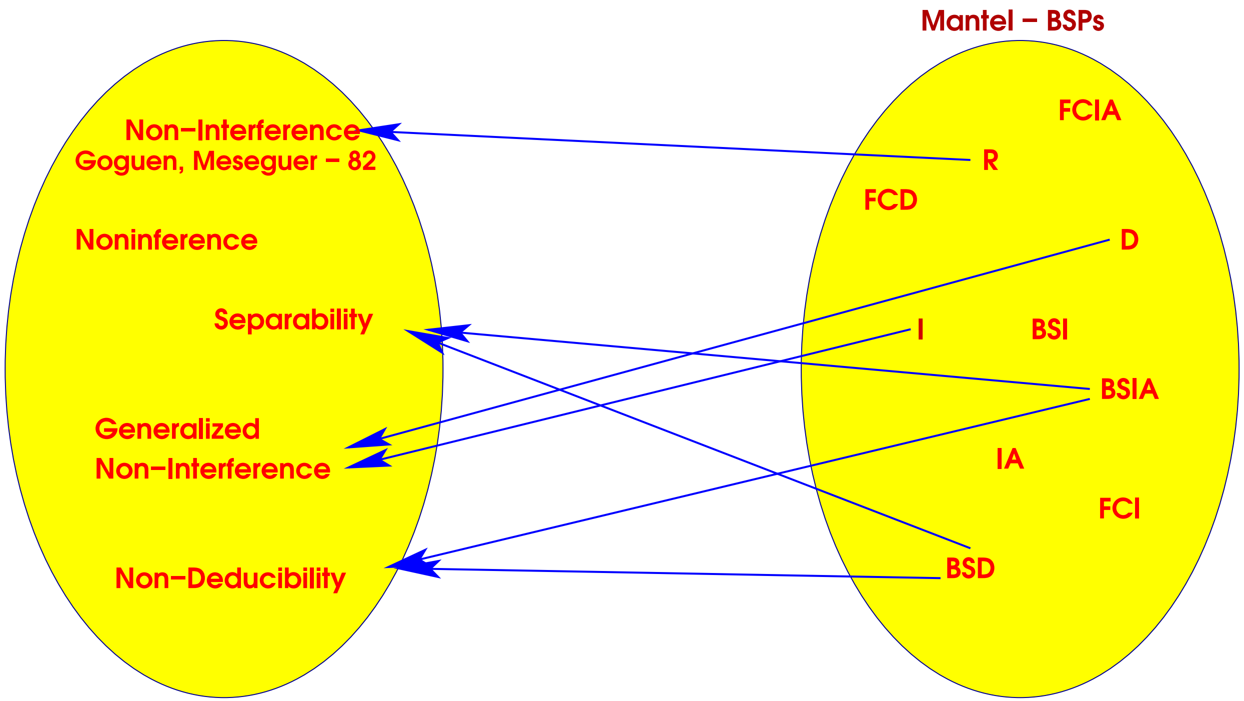


Background

Granting, restricting & controlling the flow of information

Goguen, Meseguer, ’82 - Non-Interference

"What one group of users does has no effect on what other group of users does"

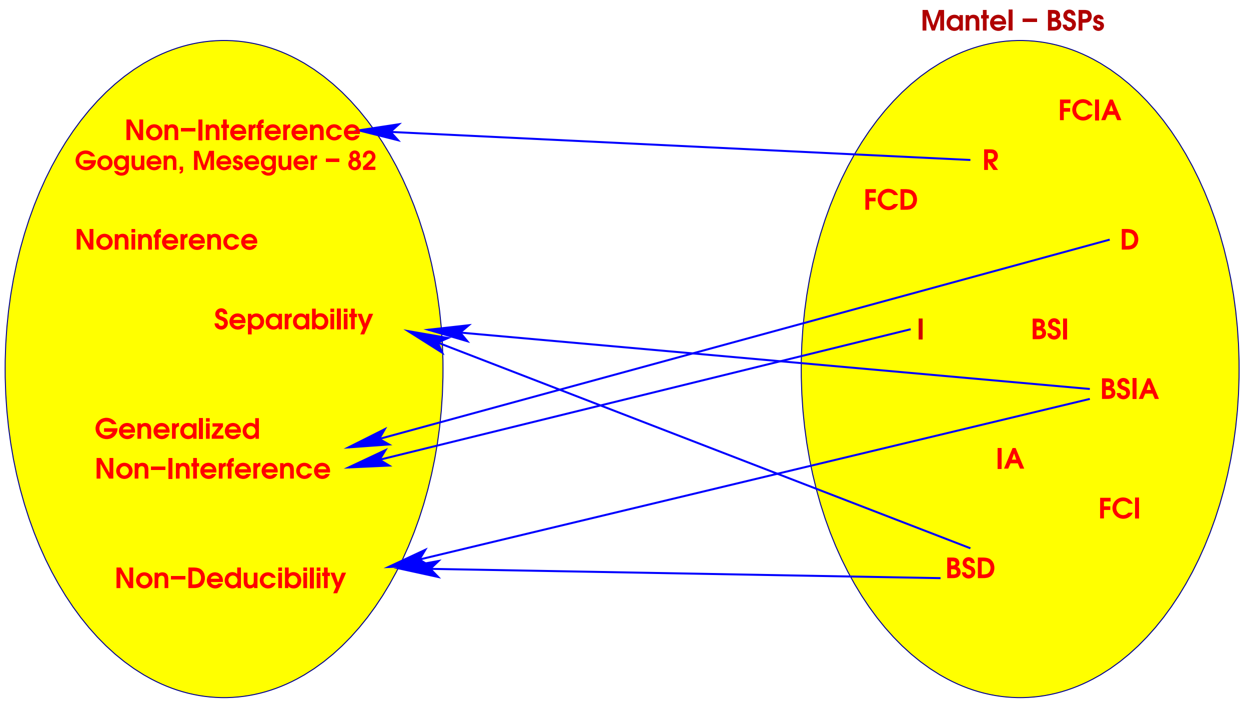


Background

Granting, restricting & controlling the flow of information

Goguen, Meseguer, ’82 - Non-Interference

"What one group of users does has no effect on what other group of users does"



Can we automate Verification of Security?

Basic Security Predicates (BSPs)

Events. Visible, Confidential, Neither

Basic Security Predicates (BSPs)

Events. Visible, Confidential, Neither

Trace: finite sequence of events

Basic Security Predicates (BSPs)

Events. Visible, Confidential, Neither

Trace: finite sequence of events

System: sets of traces

Basic Security Predicates (BSPs)

Events. Visible, Confidential, Neither

Trace: finite sequence of events

System: sets of traces

Trace based information flow properties in BSPs

Basic Security Predicates (BSPs)

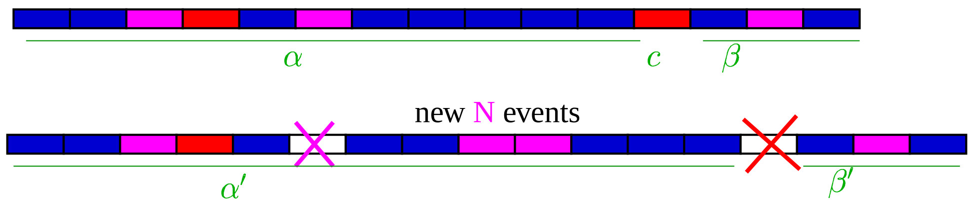
Events. Visible, Confidential, Neither

Trace: finite sequence of events

System: sets of traces

Trace based information flow properties in BSPs

BSP Deletion (D)



Basic Security Predicates (BSPs)

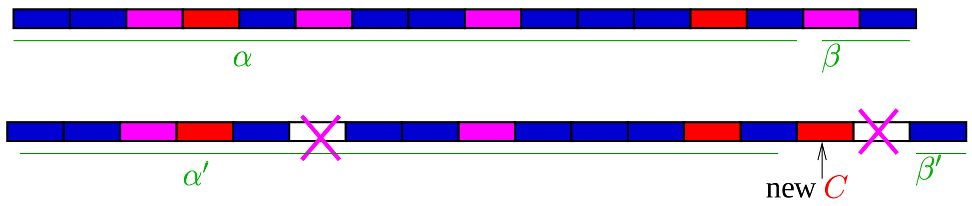
Events. Visible, Confidential, Neither

Trace: finite sequence of events

System: sets of traces

Trace based information flow properties in BSPs

BSP Insertion (I)



Basic Security Predicates (BSPs)

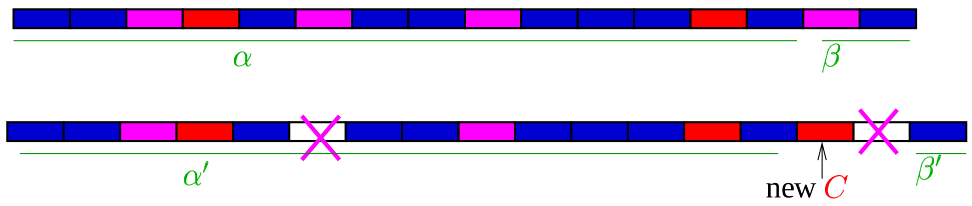
Events. Visible, Confidential, Neither

Trace: finite sequence of events

System: sets of traces

Trace based information flow properties in BSPs

BSP Insert X-Admissible (IA^X)



Basic Security Predicates (BSPs)

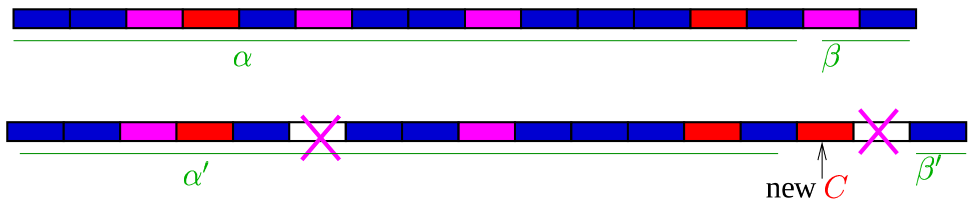
Events. Visible, Confidential, Neither

Trace: finite sequence of events

System: sets of traces

Trace based information flow properties in BSPs

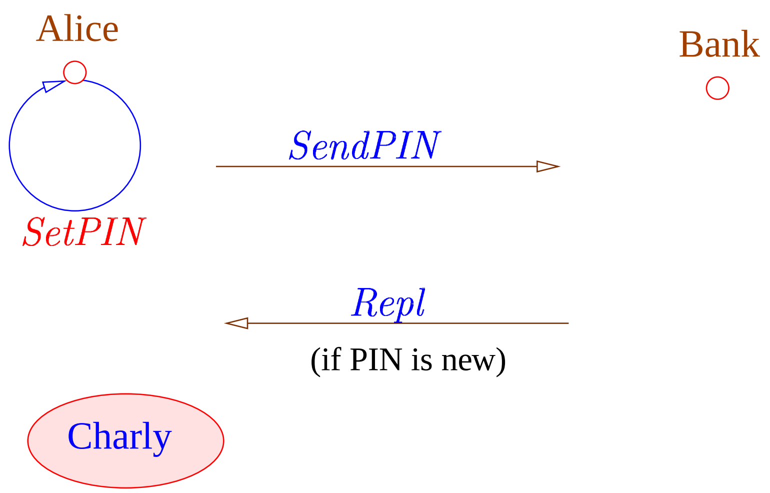
BSP Insert X-Admissible (IA^X)



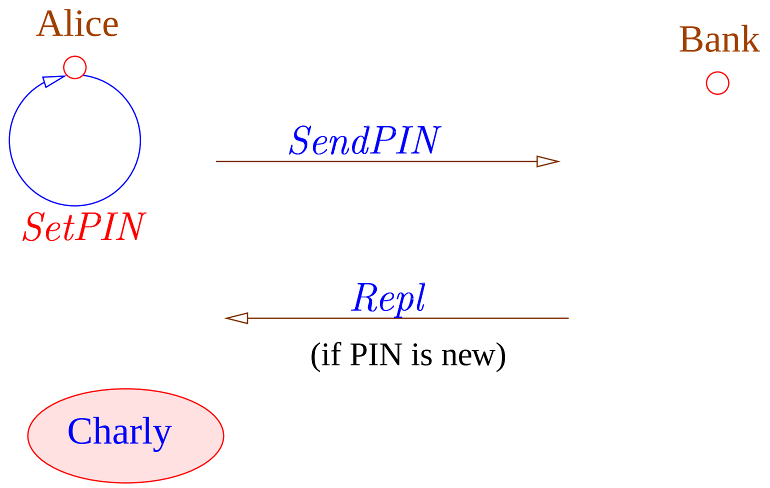
Generalized Non-Interference - I and D

Noninference - R

An Example

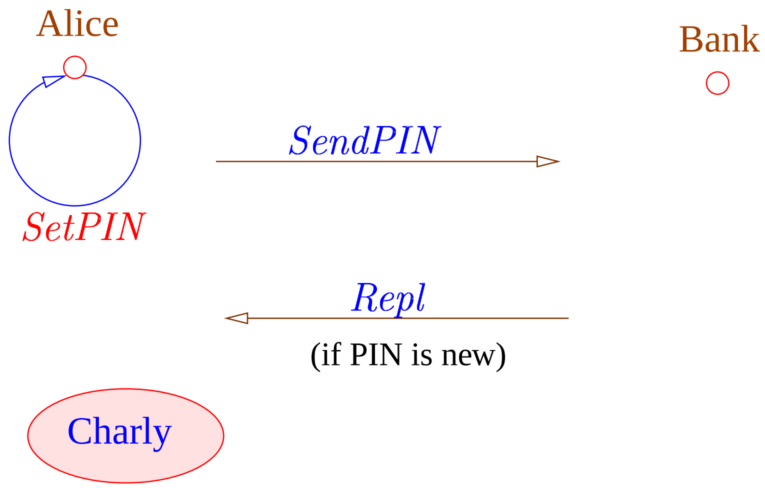


An Example



$V = \{SendPIN, Repl\}$
$C = \{SetPIN\}$
$N = \phi$

An Example



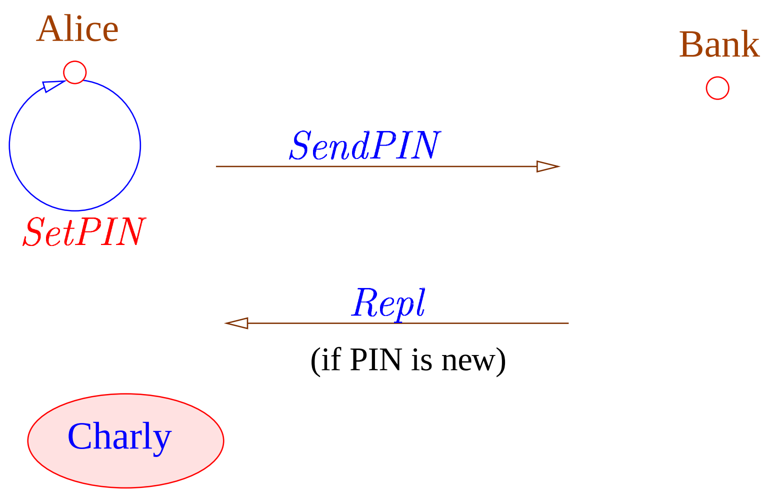
$V = \{SendPIN, Repl\}$

$C = \{SetPIN\}$

$N = \phi$

Tr = { $SetPIN\ SendPIN\ Repl$,
$SendPIN$ } + prefixes

An Example

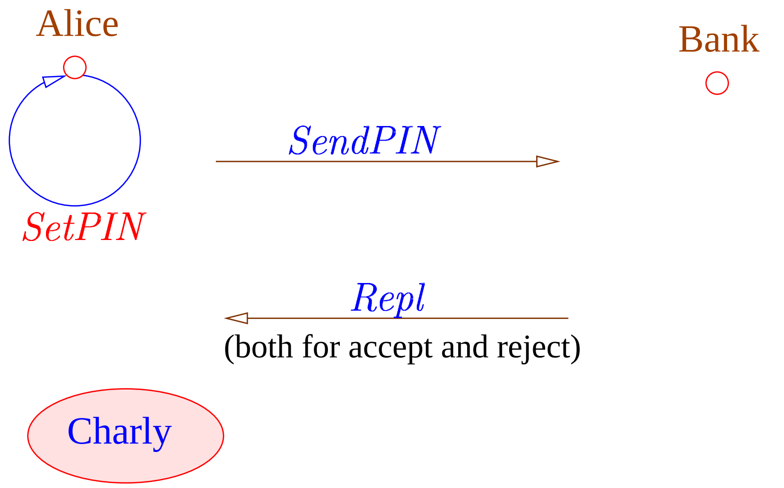


$V = \{SendPIN, Repl\}$
$C = \{SetPIN\}$
$N = \phi$

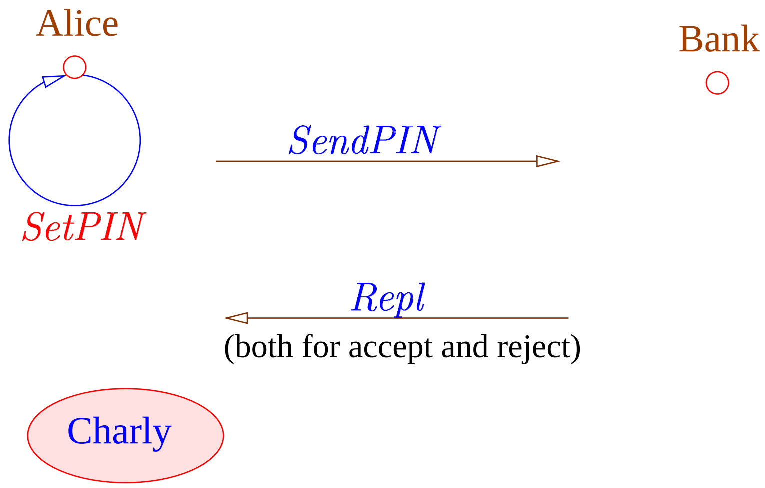
Tr = { $SetPIN$ $SendPIN$ $Repl$,
$SendPIN$ } + prefixes

Confidentiality compromised. BSP Deletion fails

Example ..contd



Example ..contd

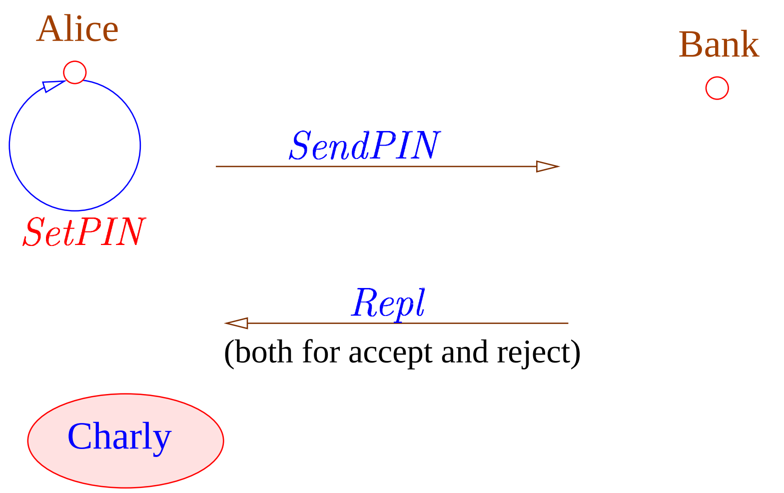


$V = \{SendPIN, Repl\}$

$C = \{SetPIN\}$

$N = \phi$

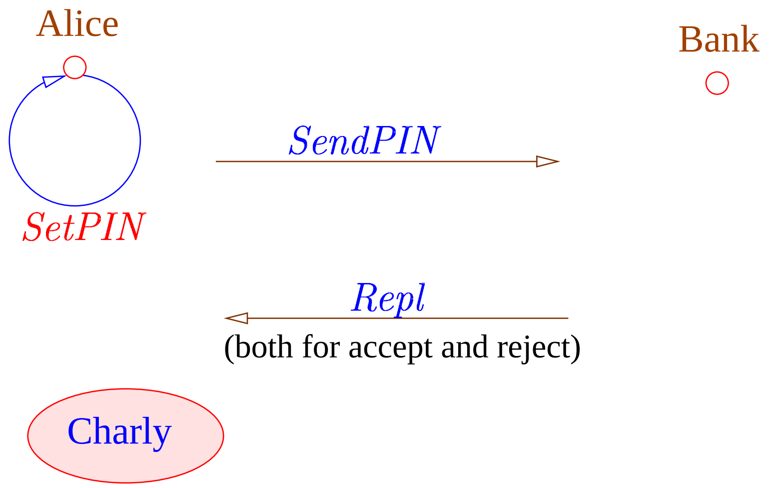
Example ..contd



$V = \{SendPIN, Repl\}$
$C = \{SetPIN\}$
$N = \phi$

Tr = { $SetPIN$ $SendPIN$ $Repl$,
$SendPIN$ $Repl$ } + prefixes

Example ..contd



$V = \{SendPIN, Repl\}$
$C = \{SetPIN\}$
$N = \phi$

Tr = { $SetPIN\ SendPIN\ Repl$,
$SendPIN\ Repl$ } + prefixes

Confidentiality maintained. BSP Deletion holds

Verification

Can we automate the verification of security properties?

Verification

Can we automate the verification of security properties?

Properties of sets of traces, Classical Model-checking techniques (Temporal Logic etc) cannot be used

Verification

Can we automate the verification of security properties?

Properties of sets of traces, Classical Model-checking techniques (Temporal Logic etc) cannot be used

Unwinding verification technique - Not Complete

Verification

Can we automate the verification of security properties?

Properties of sets of traces, Classical Model-checking techniques (Temporal Logic etc) cannot be used

Unwinding verification technique - Not Complete

Good News. For finite systems, Yes

Verification

Can we automate the verification of security properties?

Properties of sets of traces, Classical Model-checking techniques (Temporal Logic etc) cannot be used

Unwinding verification technique - Not Complete

Good News. For finite systems, Yes

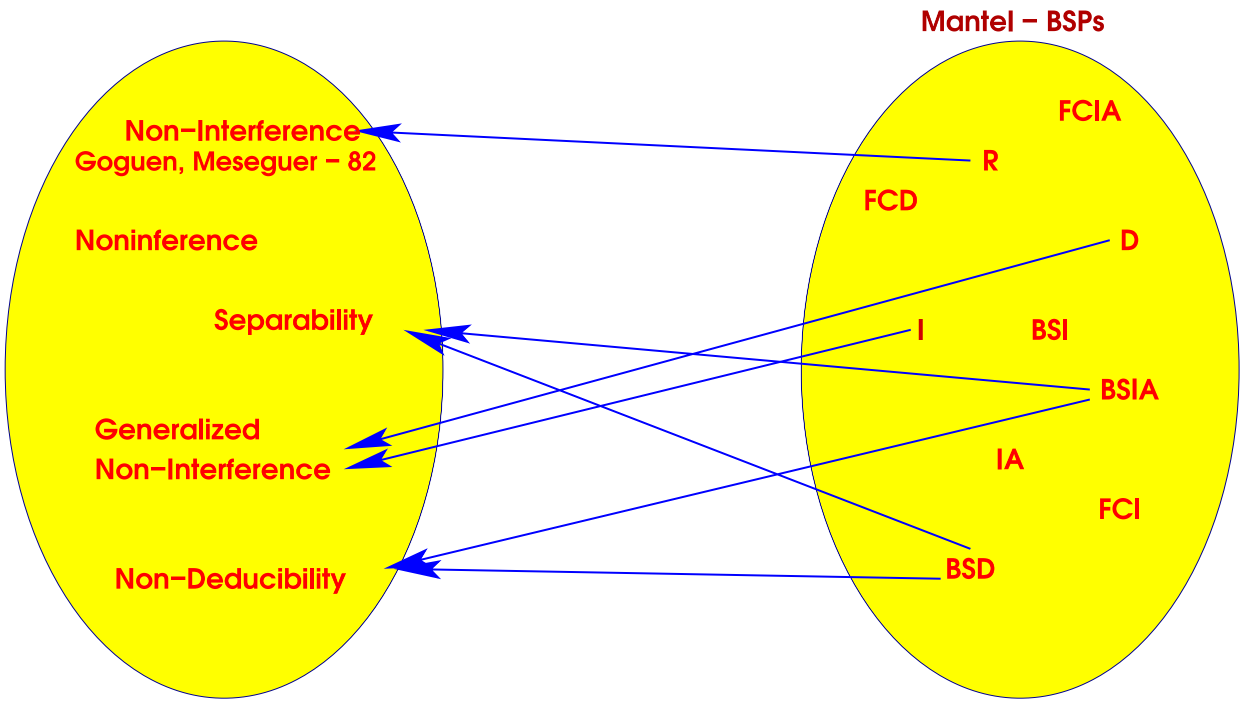
Automated Verification technique - BSP on Finite State Automaton

Language-theoretic Operations

L be a language over Σ, X subset of Σ

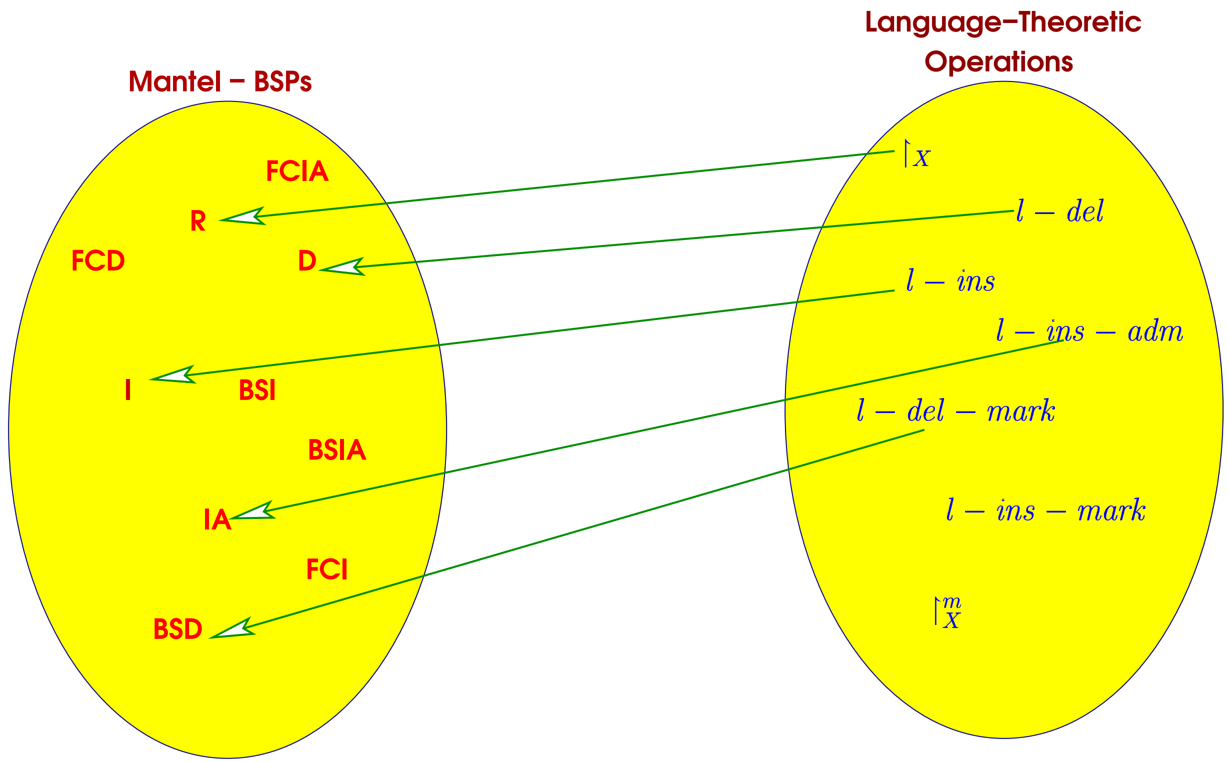
Language-theoretic Operations

L be a language over Σ, X subset of Σ



Language-theoretic Operations

L be a language over Σ, X subset of Σ



Language-theoretic Operations

L be a language over Σ, X subset of Σ

$L\upharpoonright_X := \{\tau\upharpoonright_X \mid \tau \text{ in } L\}$,
$\tau\upharpoonright_X$, deletes events that are not elements of X

Language-theoretic Operations

L be a language over Σ, X subset of Σ

$L\upharpoonright_X := \{\tau\upharpoonright_X \mid \tau$ in $L\}$,
$\tau\upharpoonright_X$, deletes events that are not elements of X

l-del$(L) := \{\alpha\beta \mid \alpha c\beta$ in L, no C events in $\beta\}$
deletes the last occuring C-event

Language-theoretic Operations

L be a language over Σ, X subset of Σ

$L\restriction_X := \{\tau\restriction_X \mid \tau$ in $L\}$,
$\tau\restriction_X$, deletes events that are not elements of X

l-del$(L) := \{\alpha\beta \mid \alpha c\beta$ in L, no C events in $\beta\}$
deletes the last occuring C-event

l-ins$(L) := \{\alpha c\beta \mid \alpha\beta$ in L, no C events in $\beta\}$
inserts a C-event in a position after which no C-events occur

Language-theoretic Operations

L be a language over Σ, X subset of Σ

$L\upharpoonright_X := \{\tau\upharpoonright_X \mid \tau$ in $L\}$,
$\tau\upharpoonright_X$, deletes events that are not elements of X

l-del$(L) := \{\alpha\beta \mid \alpha c\beta$ in L, no C events in $\beta\}$
deletes the last occuring C-event

l-ins$(L) := \{\alpha c\beta \mid \alpha\beta$ in L, no C events in $\beta\}$
inserts a C-event in a position after which no C-events occur

l-ins-adm$^X(L) := \{\alpha c\beta \mid \alpha\beta$ in L, no C events in β, there exists γc in L, $\gamma =_{\bar{X}} \alpha\}$

Language Inclusion Problem

"L satisfies a BSP P" is reduced to "$op_1(L) \subseteq op_2(L)$"

Language Inclusion Problem

"L satisfies a BSP P" is reduced to "$op_1(L) \subseteq op_2(L)$"

- Removal R iff $L\upharpoonright_V \subseteq_N L$.
- Deletion D iff *l-del*$(L) \subseteq_N L$.
- Insertion I iff *l-ins*$(L) \subseteq_N L$.
- Strict Removal SR iff $L\upharpoonright_{\overline{C}} \subseteq L$.
- Strict Deletion SD iff *l-del*$(L) \subseteq L$.

Language Inclusion Problem for BSP D

L satisfies BSP D

l-del$(L) \subseteq_{N} L$

Language Inclusion Problem for BSP D

L satisfies BSP D

Any τ in *l-del*(L)

l-del$(L) \subseteq_N L$

Language Inclusion Problem for BSP D

L satisfies BSP D

Any τ in *l-del*(L)

$\tau - \alpha\beta$, no C events in β, $\alpha c\beta$ in L

l-del$(L) \subseteq_N L$

Language Inclusion Problem for BSP D

L satisfies BSP D

Any τ in *l-del*(L)

$\tau - \alpha\beta$, no C events in β, $\alpha c \beta$ in L

Since L sat D, there exists $\tau' = \alpha'\beta'$ in L such that $\alpha =_N \alpha'$ and $\beta =_N \beta'$

l-del$(L) \subseteq_N L$

Language Inclusion Problem for BSP D

L satisfies BSP D

Any τ in *l-del*(L)

$\tau - \alpha\beta$, no C events in β, $\alpha c \beta$ in L

Since L sat D, there exists $\tau' = \alpha'\beta'$ in L such that $\alpha =_N \alpha'$ and $\beta =_N \beta'$

τ' equivalent to τ modulo N-corrections

l-del$(L) \subseteq_N L$

Language Inclusion Problem for BSP D

l-del$(L) \subseteq_N L$

L satisfies BSP D

Language Inclusion Problem for BSP D

I-del$(L) \subseteq_N L$

Any τ of form $\alpha c \beta$ in L, no C-events in β

L satisfies BSP D

Language Inclusion Problem for BSP D

l-del$(L) \subseteq_N L$

Any τ of form $\alpha c \beta$ in L, no C-events in β

$\alpha\beta$ belongs to *l-del*(L)

L satisfies BSP D

Language Inclusion Problem for BSP D

l-del$(L) \subseteq_N L$

Any τ of form $\alpha c \beta$ in L, no C-events in β

$\alpha\beta$ belongs to *l-del*(L)

Since *l-del*$(L) \subseteq_N L$, there exists $\tau' =_N \tau$

L satisfies BSP D

Language Inclusion Problem for BSP D

l-del$(L) \subseteq_N L$

Any τ of form $\alpha c \beta$ in L, no C-events in β

$\alpha\beta$ belongs to *l-del*(L)

Since *l-del*$(L) \subseteq_N L$, there exists $\tau' =_N \tau$

τ' as $\alpha'\beta'$

L satisfies BSP D

Regularity Preservation

How to automate the checking of Language Inclusion?

Regularity Preservation

How to automate the checking of Language Inclusion?

$L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$

Regularity Preservation

How to automate the checking of Language Inclusion?

$L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$

Given automata for L, algorithm for constructing automata for op(L)?

Regularity Preservation

How to automate the checking of Language Inclusion?

$L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$

Given automata for L, algorithm for constructing automata for op(L)?

$L \upharpoonright_X$

by replacing transitions $p \xrightarrow{a} q$, with $a \notin X$, in $\mathcal{A}$, by an ϵ-transition $p \xrightarrow{\epsilon} q$

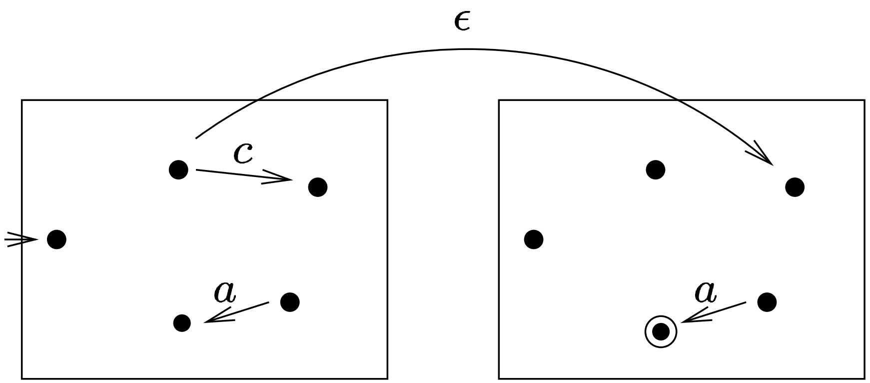
Regularity Preservation

How to automate the checking of Language Inclusion?

$L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$

Given automata for L, algorithm for constructing automata for op(L)?

l-del(L)



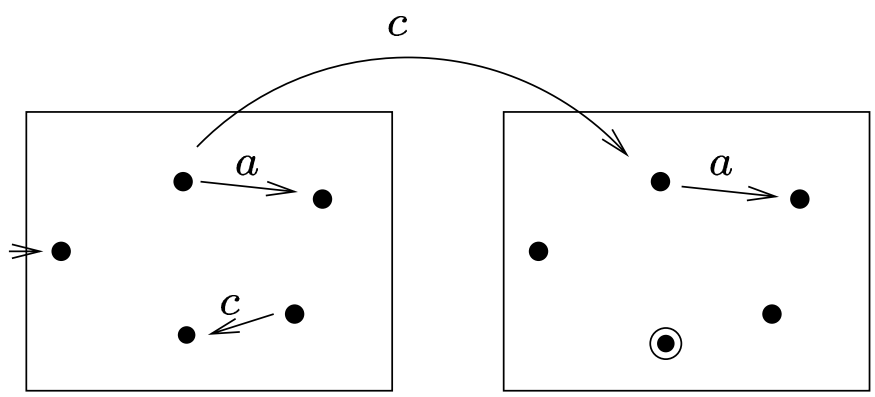
Regularity Preservation

How to automate the checking of Language Inclusion?

$L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$

Given automata for L, algorithm for constructing automata for op(L)?

l-ins(L)



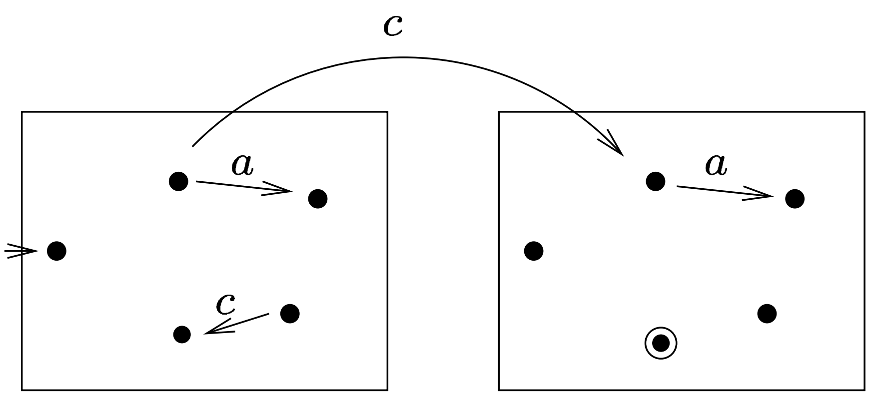
Regularity Preservation

How to automate the checking of Language Inclusion?

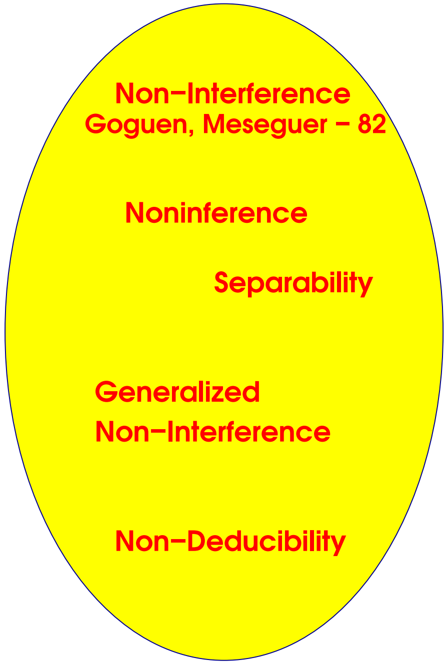
$L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$

Given automata for L, algorithm for constructing automata for op(L)?

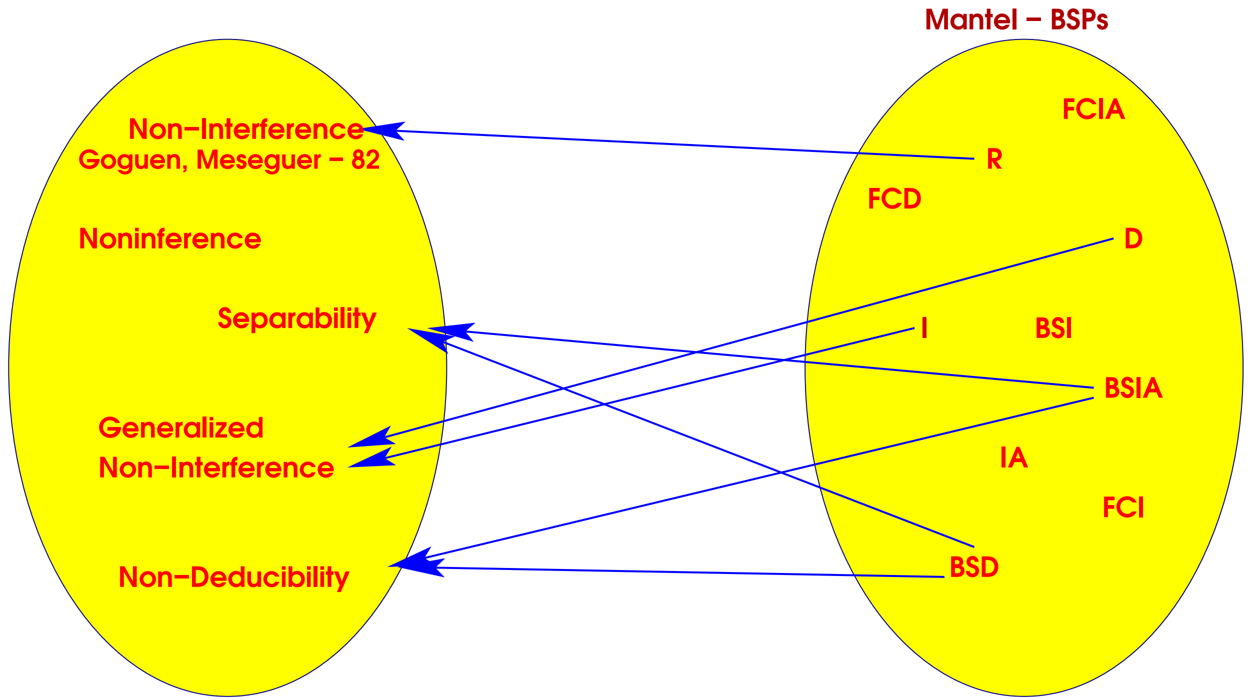
l-ins-adm$^X(L)$



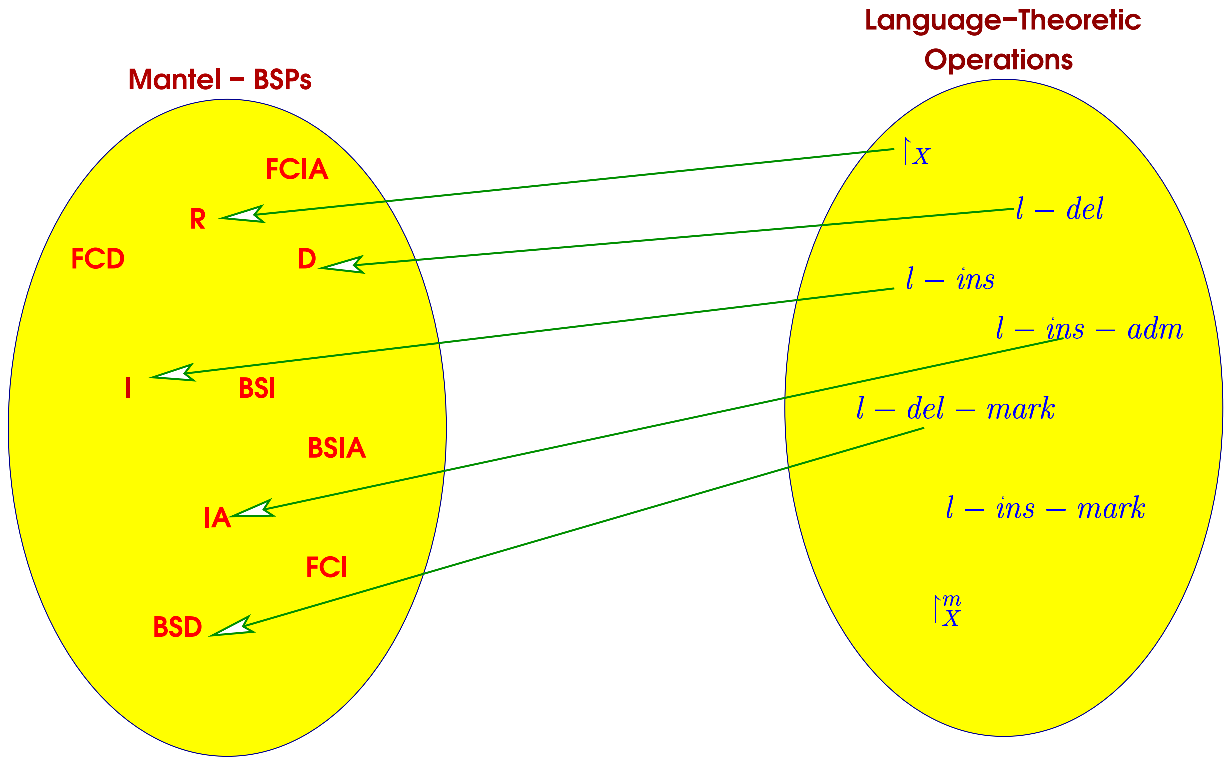
To sum up


Non−Interference
Goguen, Meseguer − 82
Noninference
Separability
Generalized
Non−Interference
Non−Deducibility

To sum up


Mantel – BSPs
Non–Interference
Goguen, Meseguer – 82
Noninference
Separability
Generalized
Non–Interference
Non–Deducibility
FCIA
R
FCD
D
I
BSI
BSIA
IA
FCI
BSD

To sum up



Mantel – BSPs
FCIA
R
FCD
D
I
BSI
BSIA
IA
FCI
BSD
Language–Theoretic Operations
$\upharpoonright_X$
$l-del$
$l-ins$
$l-ins-adm$
$l-del-mark$
$l-ins-mark$
$\upharpoonright_X^m$

Conclusion

Running time is exponential in the number of states of the given finite state transition system $2^{O(n)}$

Conclusion

Running time is exponential in the number of states of the given finite state transition system $2^{O(n)}$

Sound and Complete characterisation of Security properties in terms of Language-theoretic Operations

Conclusion

Running time is exponential in the number of states of the given finite state transition system $2^{O(n)}$

Sound and Complete characterisation of Security properties in terms of Language-theoretic Operations

Automatically verify trace based information flow properties for finite state systems

Conclusion

Running time is exponential in the number of states of the given finite state transition system $2^{O(n)}$

Sound and Complete characterisation of Security properties in terms of Language-theoretic Operations

Automatically verify trace based information flow properties for finite state systems

For infinite state systems?

Thank You