# Synthesis, Analysis, and Verification

Lecture 08

WS1S: Automata-Based Decision Procedure

Tree Automata and WSkS

Lectures:

**Viktor Kuncak**

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

# Weak Monadic Second-Order Logic of One Successor

$$F ::= v \subseteq v \mid succ(v, v) \mid F \vee F \mid \neg F \mid \exists v.F$$

succ ( {k}, {k+1} )

finite subset of {0, 1, 2, 3, ... }

$A, B \subseteq \{0,1,2,3,\ldots\}$ $\quad$ $A, B$ - finite

$$A \subseteq B$$

$$\begin{array}{c|ccccccc|l}
 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & \\
A & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0^\omega = \{1,2,3,5\} \\
B & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0^\omega = \{1,2,3,4,5\}
\end{array}$$

$\Downarrow A$

alphabet $\Sigma = \left\{ \boxed{\begin{smallmatrix}0\\0\end{smallmatrix}}, \boxed{\begin{smallmatrix}0\\1\end{smallmatrix}}, \boxed{\begin{smallmatrix}1\\0\end{smallmatrix}}, \boxed{\begin{smallmatrix}1\\1\end{smallmatrix}} \right\} \approx \{0,1\} \times \{0,1\}$

$\approx \{\text{"A"}, \text{"B"}\} \to \{0,1\}$

$\Sigma = \text{Vars} \to \{0,1\}$

A: B: $\binom{0}{0}, \binom{0}{1}, \binom{1}{1}$

M:


$w : \binom{0}{0} \binom{1}{1} \binom{1}{1} \binom{1}{1} \binom{0}{1} \binom{1}{1} \binom{0}{0}$

$\alpha_A(w) = \{1,2,3,5\}$ $\qquad$ $\alpha(w)(A) = \alpha_A(w)$

$\alpha_B(w) = \{1,2,3,4,5\}$ $\qquad$ $\alpha(w)(B) = \alpha_B(w)$

$A \subseteq B$

$$w \in L(M) \iff \alpha_A(w) \subseteq \alpha_B(w)$$

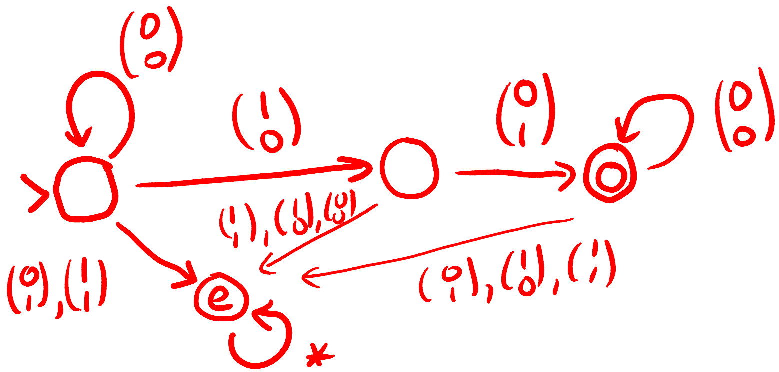
$$w \in L(M(F)) \iff w \models F \overset{\text{def}}{\iff} \alpha(w) \models F$$

emptiness testing

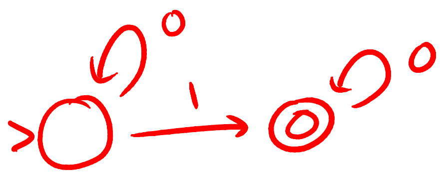
succ(X, Y)

$\{k\}, \{k+1\}$

| | | | | | | |
|---|---|---|---|---|---|---|
| X | 0 | 0 | 0 | 1 | 0 | 0 |
| Y | 0 | 0 | 0 | 0 | 1 | 0 |



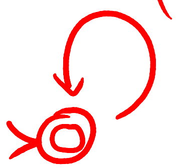
X has exactly one element



$P = Q \cup R$

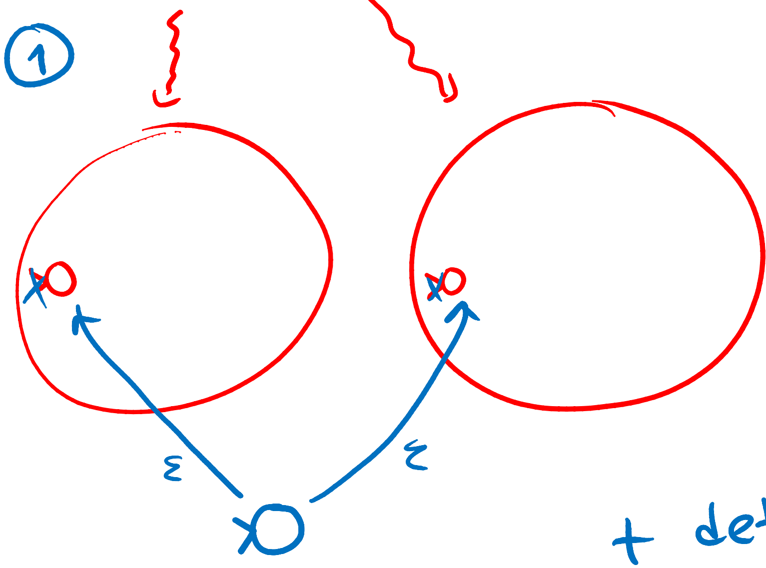
$$\begin{array}{c} P \\ Q \\ R \end{array} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$



$F_1 \vee F_2$

1



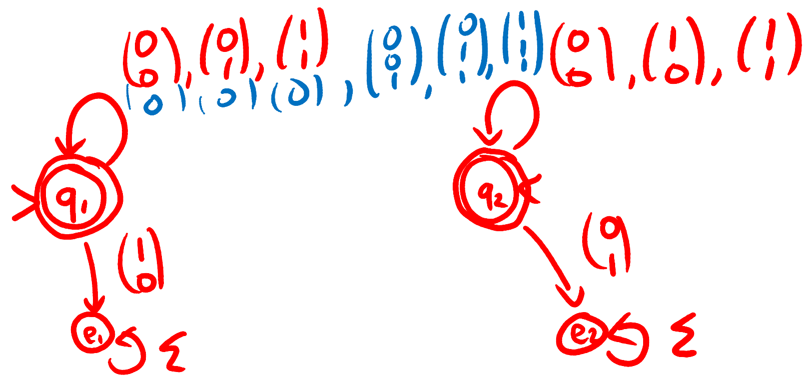

+ determinize

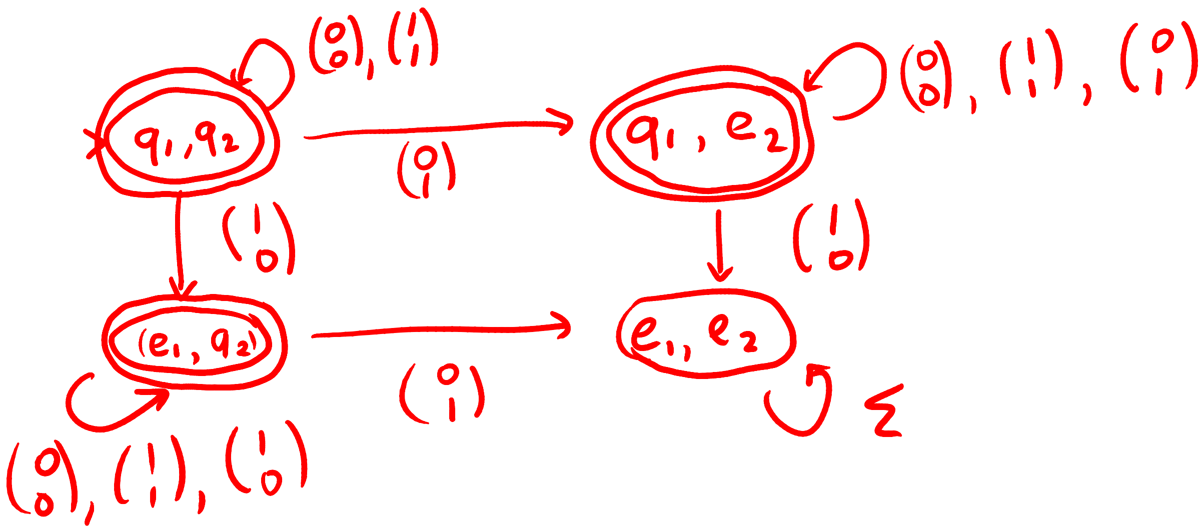
2

$$F_1 \to (Q_1, \delta_1, f_1, q_1)$$
$$F_2 \to (Q_2, \delta_2, f_2, q_2)$$

$$\big(Q_1 \times Q,\ \delta_1 * \delta_2,\ f_1 * f_2,\ (q_1, q_2)\big)$$

Create automaton for $A \subseteq B \vee B \subseteq A$

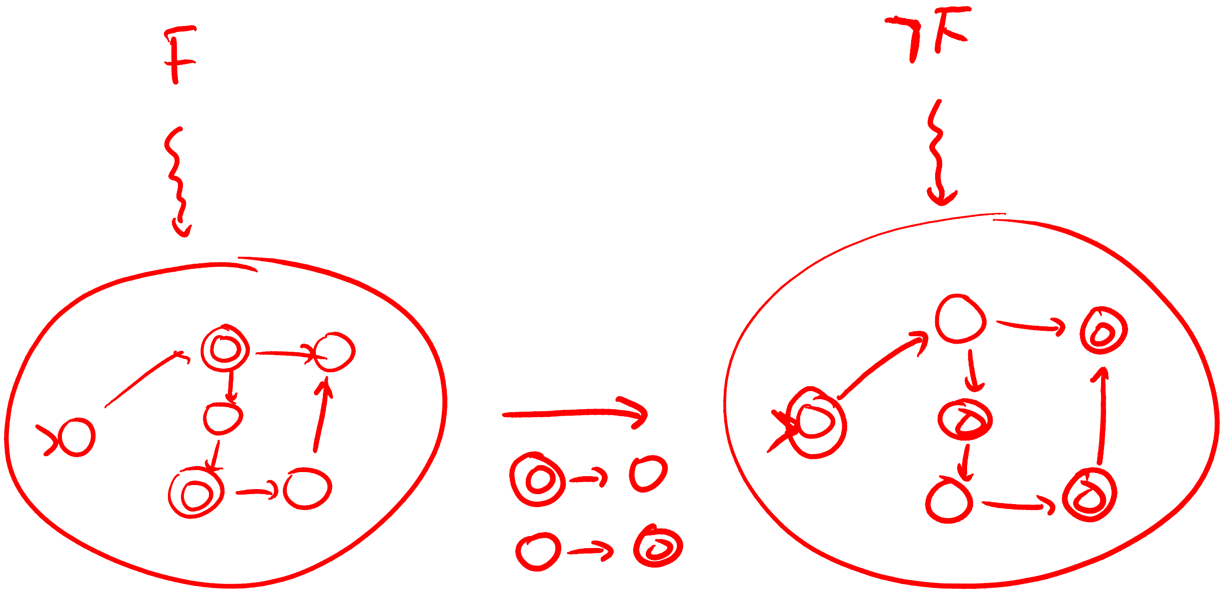



How about: $A \subseteq B \vee B \subseteq C$

$$\begin{array}{c} A \\ B \\ C \end{array} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$\neg F$

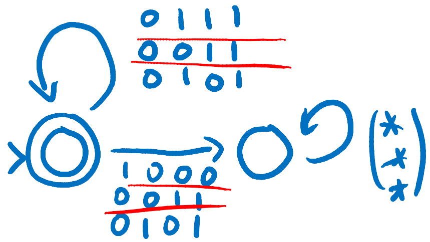


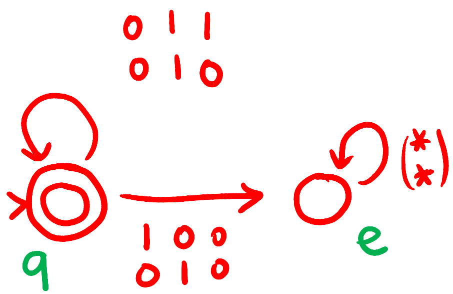
$\exists X. F$

$\exists X. X \subseteq A$

$$\exists X. \quad \underbrace{P = X \cup Q}$$

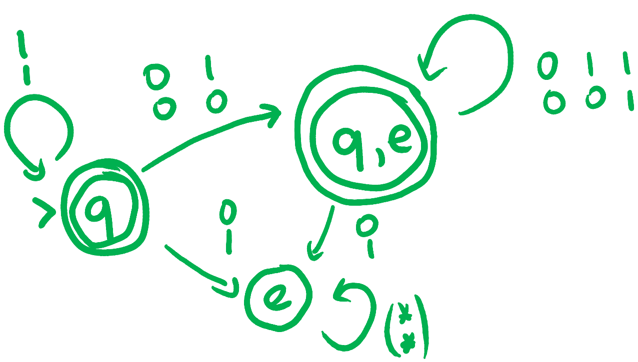
$$\Sigma_3 : \begin{pmatrix} P \\ X \\ Q \end{pmatrix}$$





$$\Sigma_2 : \begin{pmatrix} P \\ Q \end{pmatrix}$$

nondeterministic - determinize it



$$\delta(\{q_1, q_2, \ldots, q_n\}, a)$$

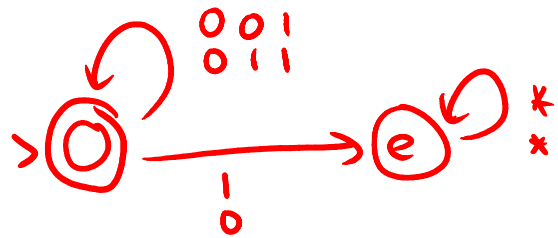
$$= \{\delta(q_1, a), \ldots, \delta(q_n, a)\}$$

$$Q \subseteq P$$

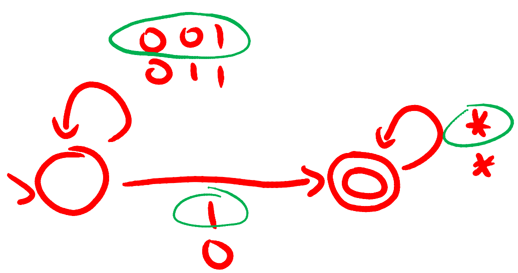
$\exists X.\ \neg(X \subseteq Y)$

$\begin{pmatrix} X \\ Y \end{pmatrix}$

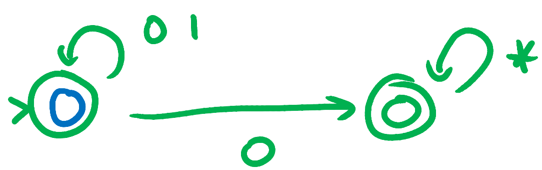
$X \subseteq Y$:



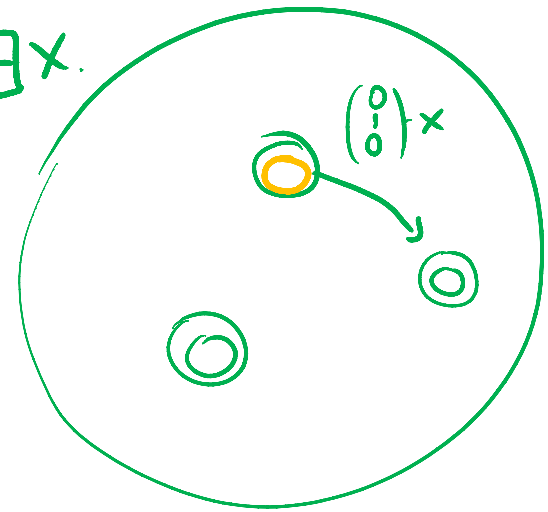

$\neg(X \subseteq Y)$:




project:




$\exists X.$




$$\begin{array}{c} Y_1 \\ \vdots \\ Y_k \\ X \\ Y_{k+1} \\ \vdots \\ Y_n \end{array} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ * \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

# WS1S As Extension of Presburger Arithmetic

Finite set of natural numbers S can represent bits in binary representation

$$N(S) = \sum_{i \in S} 2^i$$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $i$: | 0 | 1 | 2 | 3 | 4 | 5 | |
| $2^i$: | 1 | 2 | 4 | 8 | 16 | 32 | |
| S: | 0 | 1 | 1 | 1 | 0 | 0 | $= \{1,2,3\}$ |
| N(S): | | 2+4+8 | | | | | $= 14$ |

succ(X, Y)    X, Y are powers of two and $Y = 2X$

singleton    power of two

$S_1 \cup S_2$    bitwise or

$S_1 \cap S_2$    bitwise and

We can interpret sets as numbers

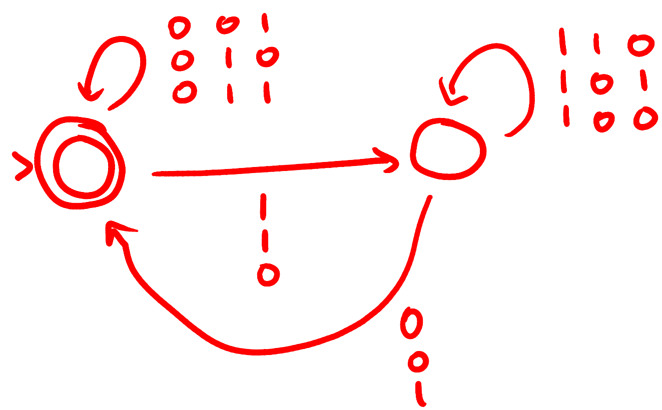
What do we gain?

# WS1S as PA + Bitwise Operators

- Non-negative integer constants and variables
- Boolean operators ($\wedge, \vee, \neg$)
- Linear arithmetic operator ($+,\ c \cdot x$)
- Bitwise operators (|, &, !)
- Quantifiers over numbers and bit positions

$Z = X + Y$

$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$



# WS1S as PA + Bitwise Operators

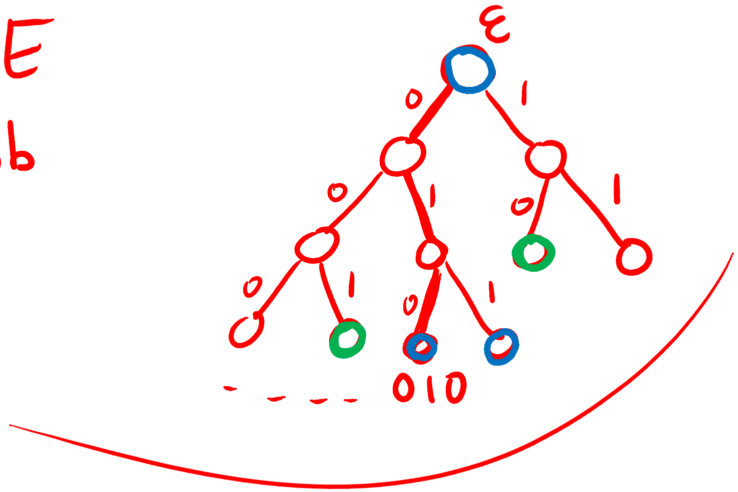
$$
\begin{array}{rcl}
F & ::= & F \wedge F \mid F \vee F \mid \neg F \mid t_N < t_N \mid t_N = t_N \\
& \mid & t_N[t_P] \mid t_P < t_P \mid t_P = t_P \mid (C \mid t_N) \mid t_N \underset{t_P}{\sim} t_N \\
& \mid & \forall_{\mathsf{pos}} k.F \mid \exists_{\mathsf{pos}} k.F \mid \forall x.F \mid \exists x.F \\
t_N & ::= & x \mid C \mid t_N + t_N \mid C \cdot t_N \mid t_N \,\mathsf{div}\, C \mid t_N \,\%\, C \\
& \mid & (t_N \veebar t_N) \mid (t_N \barwedge t_N) \mid t_N \ll C \mid t_N \gg C \\
& \mid & 2^{t_P} \mid t_N[t_P..^{+}C] \mid t_N[0..t_P] \\
t_P & ::= & k \mid C \mid k + C \mid k \dot{-} C \mid \mathsf{maxBit}(t_N) \\
C & ::= & \text{non-negative integer constant}
\end{array}
$$

Fig. 8. Syntax of WS1S where sets denote natural numbers ($T_N$) and elements denote positions ($T_P$) in binary representations of numbers

# Infinite Binary Tree in WS2S

PALE

Jahob




$succ_0(\{w\}, \{w0\})$

$succ_0(w) = w0$

$succ_1(w) = w1$

number: was string in unary alphabet

4 is 1111

now: tree node: string in binary alphabet

root . left . right . left

is: 0 1 0

sets of nodes:

$A = \{\varepsilon, 010, 011\}$

$B = \{001, 10\}$

Tree automaton: run on a tree, accept labeling of trees

top-down bottom-up