Recapitulation:

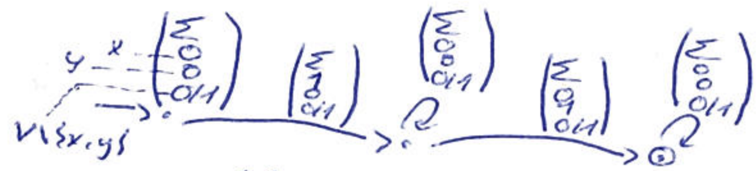
- Given a set of variables V that contains the free variables of φ.
- Construct automaton A_φ over Σ_V
- Makes use of extended alphabet $\Sigma_V := \Sigma \times \{0,1\}^V$
  ↳ Allows us to encode interpretations I into word w_I over Σ_V.

Goal:

$w_I \in \mathcal{L}(A_\varphi)$ iff $S_w, I \models \varphi$.

Base Cases:

$A_{x<y}$ over Σ_V:



- Σ split into several transitions, one for each letter
- there may be further free variables in $V \setminus \{x,y\}$.

Consider

$\exists x: \exists y: \exists z: (P_a(x) \wedge P_b(y) \wedge \underbrace{x<y}_{} \wedge y<z)$

$A_{x<y}$ will use letters over $\Sigma \times \{0,1\}^{\{x,y,z\}}$

Induction step:

$A_{\exists X:\varphi}$ over Σ_V: · Construct A_φ over $\Sigma_{V \cup \{X\}}$.

Let it be

$A_\varphi = (Q, q_0, \rightarrow, Q_F)$.

Define

$A_{\exists X:\varphi} := (Q, q_0, \rightarrow', Q_F)$ over Σ_V

by projecting away X:

$q \xrightarrow{a'}{}' q'$ if $q \xrightarrow{a} q'$ and $a' = a|_V$.

Show that $w_I \in \mathcal{L}(A_{\exists X:\varphi})$ iff $S_w, I \models \exists X:\varphi$

"$\Rightarrow$" Let $w_I \in \mathcal{L}(A_{\exists X:\varphi})$ with accepting run

$$w_I = \quad q_0 \xrightarrow{\begin{pmatrix} a_0 \\ c_0^1 \\ \vdots \\ c_0^n \end{pmatrix}} q_1 \xrightarrow{\begin{pmatrix} a_1 \\ c_1^1 \\ \vdots \\ c_1^n \end{pmatrix}} q_2 \to \dots \to q_f \in Q_F \text{ of } A_{\exists X:\varphi}.$$

As the paths coincide for $A_{\exists X:\varphi}$ and A_φ,
the transition sequence also exists in A_φ:

$$X \quad q_0 \xrightarrow{\begin{pmatrix} a_0 \\ c_0^1 \\ \vdots \\ c_0^n \\ \hline c_0^{n+1} \end{pmatrix}} q_1 \xrightarrow{\begin{pmatrix} a_1 \\ c_1^1 \\ \vdots \\ c_1^n \\ \hline c_1^{n+1} \end{pmatrix}} q_2 \to \dots \to q_f \in Q_F \text{ of } A_\varphi.$$

It yields an interpretation for X:
↳ Those positions k with $c_k^{n+1} = 1$
↳ Let this set of positions be M.

The resulting word is $w_{I[M/X]} \in \mathcal{L}(A_\varphi)$.
By the hypothesis,

$$S_w, I[M/X] \models \varphi.$$

This means, $S_w, I \models \exists X:\varphi$.

"$\Leftarrow$" Let $S_w, I \models \exists X:\varphi$
This means there is $M \subseteq D_w$ so that

$$S_w, I[M/X] \models \varphi$$

By the hypothesis,

$$w_{I[M/X]} \in \mathcal{L}(A_\varphi).$$

This means there is an accepting run of A_φ on $w_{I[M/X]}$.
Thus, there is an accepting run of $A_{\exists X:\varphi}$ on w_I

$\square$

Before we continue with an example,
recall construction for intersection of regular languages:

<u>Lemma:</u>

Let A, B two NFAs. Then there is an NFA $A \| B$ (sometimes also denoted by $A \times B$) that satisfies $\mathcal{L}(A \| B) = \mathcal{L}(A) \cap \mathcal{L}(B)$.

<u>Construction:</u>

Let $A = (Q, q_0, \to, Q_F)$ and $B = (\widehat{Q}, \widehat{q}_0, \widehat{\to}, \widehat{Q}_F)$.
Then

$$A \| B := (Q \times \widehat{Q}, (q_0, \widehat{q}_0), \to', Q_F \times \widehat{Q}_F),$$

where

$$(q, \widehat{q}) \xrightarrow{a}' (q', \widehat{q}'), \text{ if } q \xrightarrow{a} q' \text{ \underline{and} } \widehat{q} \,\widehat{\xrightarrow{a}}\, \widehat{q}'$$

<u>Intuitively:</u>

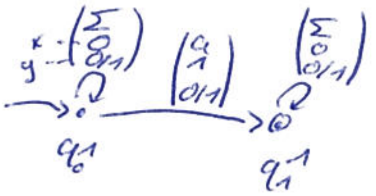
The two automata agree on their runs.

<u>Example (For automata construction out of WMSO formulas)</u>

- Let $\Sigma = \{a, b\}$
- Consider $\exists x : \exists y : P_a(x) \wedge P_b(y) \wedge x < y$
  (defines $\{a,b\}^* . a . \{a,b\}^* b . \{a,b\}^*$)

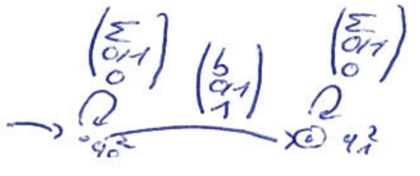
We get

$A_{P_a(x)}$ over Σ_V with $V = \{x, y\}$:



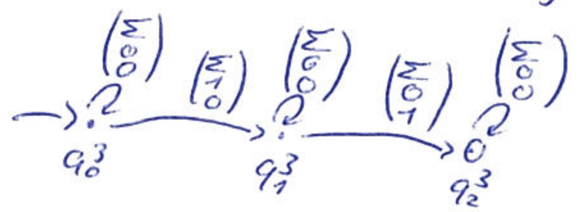
$\|$

$A_{P_b(y)}$ over Σ_V with $V = \{x, y\}$:

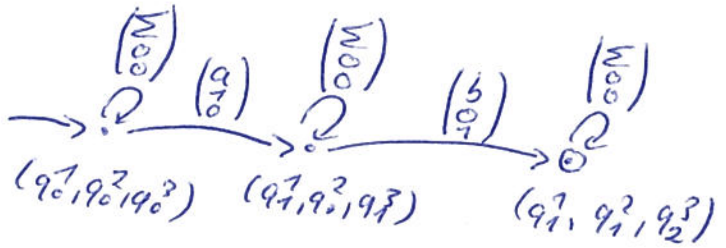


$\Downarrow$

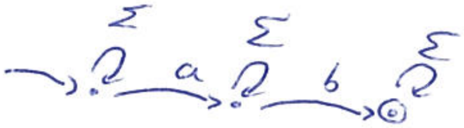
$\mathcal{A}_{x<y}$ over Σ_V with $V=\{x,y\}$:



$=$



After projection (for $\exists x$ and $\exists y$):



<u>Theorem (Büchi (I + II)):</u>

A language is regular iff it is WMSO-definable.

<u>Corollary:</u>

It is decidable, whether a WMSO-formula is valid/satisfiable.

<u>Worst-case complexity of automata construction:</u>

Let $\mathcal{A}$ and $\mathcal{B}$ have at most n states.

$\mathcal{A} \cup \mathcal{B} \rightsquigarrow 2n+1$ states

$\overline{\mathcal{A}} \rightsquigarrow 2^n$ states

$\mathcal{A}_{\exists X:\varphi} \rightsquigarrow n$ states.
(projection)

Thus, formula with k logical connectives may yield automaton of size

$$\underbrace{2^{2^{\cdot^{\cdot^{2^{c}}}}}}_{k\text{-times}}$$

Construction from left to right in Büchi's theorem gave formulas of particular shape:

$$\exists X_0 : \ldots \exists X_n : (1) \wedge \ldots \wedge (4) (\wedge (5))$$

<u>Existential WMSO</u> ($\exists$WMSO) is defined as the restriction of WMSO to formulas of the form

$$\exists X_0 : \ldots \exists X_n : \varphi$$

where φ does not contain second-order quantification.

<u>Corollary:</u>

- Every closed WMSO-formula φ has an equivalent closed formula ψ in $\exists$WMSO, where equivalent means $\mathcal{L}(\varphi) = \mathcal{L}(\psi)$.
- Thus a language is definable in WMSO iff it is definable in $\exists$WMSO.

<u>Proof:</u>

Let φ a WMSO-formula.
Construct A_φ with Büchi II.
Construct $\varphi_{\mathcal{L}(A_\varphi)}$ with Büchi I. $\square$

<u>2.3 An application: decidability of Presburger arithmetic.</u>

Presburger arithmetic: first-order logic over natural numbers with addition
↳ Fix $Sig = (\underbrace{\{0/_0, +/_2\}}_{Fun}, \underbrace{\{</_2, =/_2\}}_{Pred})$

The formulas φ and terms t in Presburger arithmetic are defined by

$t ::= 0 \mid x \mid t_1 + t_2$ $\qquad$ $\varphi ::= t_1 = t_2 \mid t_1 < t_2 \mid \varphi_1 \wedge \varphi_2 \mid \neg \varphi \mid \exists x : \varphi$

Interpret in (fixed) structure

$(\mathbb{N}, 0_{\mathbb{N}}, +_{\mathbb{N}}, <_{\mathbb{N}}, =_{\mathbb{N}})$ (usual 0, addition, less, equality)

Summary:

- Talk about 0, addition of numbers, their ordering
- Cannot use multiplication ($\leadsto$ Peano arithmetic, not even $x = y^2$).
- No second-order quantifiers (and cannot be derived).

Goal:

- Check truth of a closed formula in Presburger arithmetic.

Remark:

- Looks stronger than WMSO as you have general addition (not only suc).

Examples:

- $\text{even}(y) := \exists x : y = x + x$
- $(y = 1) := \forall x : x < y \to x = 0$
- To use constant 1 in formula φ, write

$$\exists y : (y = 1) \wedge \varphi\{y/1\} \quad \text{// Replace 1 by } y.$$

- $(x < y) := \exists z : y = x + z \wedge \neg(z = 0)$
- $(y \equiv r \bmod 5) := \exists x : r < 5 \wedge y = x + x + x + x + x + r$

Check truth by encoding to WS1S

Take WMSO without $P_a(x)$:

$$\varphi ::= \text{suc}(x, y) \mid X(x) \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \exists x : \varphi \mid \exists X : \varphi$$

Fix structure $(\mathbb{N}, \text{suc}_{\mathbb{N}})$.

$\hookrightarrow$ Second-order quantifiers still yield finite sets.

$\hookrightarrow$ But set of positions infinite

$\hookrightarrow$ Does not increase expressiveness, (see W. Thomas or M. Hofmann, M. Lange) but automata construction more complicated

Key idea:

- Binary encoding of numbers:

$42 =$	0	$+2^1$	$+0$	$+2^3$	$+0$	$+2^5$	Decimal number
	0	1	0	1	0	1	Binary number
	0	1	2	3	4	5	Positions

$42 \rightsquigarrow \{1,3,5\}$ (Set of positions)

- Replace first-order variable in Presburger arithmetic by second-order variable of WS1S.

  $\exists x \rightsquigarrow \exists X$ (set of positions that encode x in binary)

- Represent addition (already got rid of $<$):
  - $x + y = z$
  - Use second-order variable C for carry:

$$\exists C: (\forall x: first(x) \to \neg C(x)) \quad \text{// initially no carry}$$
$$\land \forall x: (\neg X(x) \land \neg Y(x) \land \neg C(x)$$
$$\to [\neg Z(x) \land (\forall y: suc(x,y) \to \neg C(y))])$$
$$\land (X(x) \land \neg Y(x) \land \neg C(x)$$
$$\to [Z(x) \land (\forall y: suc(x,y) \to \neg C(y))])$$
$$\land$$

6 more cases.

Theorem:

It is decidable whether a closed formula in Presburger arithmetic holds.

Theorem remains
- true for $(\mathbb{Z}, 0_{\mathbb{Z}}, +_{\mathbb{Z}}, =_{\mathbb{Z}}, <_{\mathbb{Z}})$
- true for $(\mathbb{R}, 0_{\mathbb{R}}, +_{\mathbb{R}}, =_{\mathbb{R}}, <_{\mathbb{R}})$
  - ↳ Reals are infinite subsets of $\mathbb{N}$.
  - ↳ Büchi automata.

- In practice: direct translation of Presburger arithmetic into finite automata
- Active field of research:
  - Decidable fragments of arithmetics
  - Complexity of automata-based decision procedures for Presburger arithmetic
    => Haase/Mehl, LIAFA
  - Regular languages with counting constraints
    => Seidl, TUM
    => Here