Proof sketch

1. If L is finite, then it is FO-definable

2. If L is co-finite, then it is FO-definable.

Proof sketch

3. If L is FO-definable (over a one-letter alphabet), then it is finite or co-finite.
   1) We define a new logic QF (quantifier-free fragment)
   2) We show that a language is QF-definable iff it is finite or co-finite
   3) We show that a language is QF-definable iff it FO-definable.

1) The logic QF

- $x < k \qquad x > k$

  $x < y + k \qquad x > y + k$

  $k < last \qquad k > last$

  are formulas for every variable x, y and every $k \geq 0$.
- If f_1, f_2 are formulas, then so are $f_1 \vee f_2$ and $f_1 \wedge f_2$

2) L is QF-definable iff it is finite or co-finite

($\rightarrow$) Let f be a sentence of QF.

Then f is an and-or combination of formulas $k < last$ and $k > last$.

- $L(k < last) = \{k+1, k+2, \ldots\}$ is co-finite (we identify words and numbers)
- $L(k > last) = \{0, 1, \ldots, k\}$ is finite
- $L(f_1 \vee f_2) = L(f_1) \cup L(f_2)$ and so if $L(f)$ and $L(g)$ finite or co-finite the L is finite or co-finite.
- $L(f_1 \wedge f_2) = L(f_1) \cap L(f_2)$ and so if $L(f)$ and $L(g)$ finite or co-finite the L is finite or co-finite.

2) L is QF-definable iff it is finite or co-finite

(←) If $L = \{k_1, \ldots, k_n\}$ is finite, then

$$(k_1 - 1 < last \wedge last < k_1 + 1) \vee \cdots \vee$$
$$(k_n - 1 < last \wedge last < k_n + 1)$$

expresses L.

If L is co-finite, then its complement is finite, and so expressed by some formula. We show that for every f some formula $neg(f)$ expresses $\overline{L(f)}$

- $neg(k < last) = (k - 1 < last \wedge last < k + 1) \vee last < k$
- $neg(f_1 \vee f_2) = neg(f_1) \wedge neg(f_2)$
- $neg(f_1 \wedge f_2) = neg(f_1) \vee neg(f_2)$

3) Every first-order formula φ has an equivalent QF-formula $QF(\varphi)$

- $QF(x < y) = x < y + 0$
- $QF(\neg\varphi) = neg\big(QF(\varphi)\big)$
- $QF(\varphi_1 \vee \varphi_2) = QF(\varphi_1) \vee QF(\varphi_2)$
- $QF(\varphi_1 \wedge \varphi_2) = QF(\varphi_1) \wedge QF(\varphi_2)$
- $QF(\exists x\, \varphi) = QF\big(\exists x\, QF(\varphi)\big)$
  - If $QF(\varphi)$ disjunction, apply $\exists x\, (\varphi_1 \vee \ldots \vee \varphi_n) = \exists x\, \varphi_1 \vee \ldots \vee \exists x\, \varphi_n$
  - If $QF(\varphi)$ conjunction (or atomic formula), see example in the next slide.

- Consider the formula

$$\exists x \quad x < y + 3 \quad \wedge$$
$$z < x + 4 \quad \wedge$$
$$z < y + 2 \quad \wedge$$
$$y < x + 1$$

- The equivalent QF-formula is

$$z < y + 8 \;\wedge\; y < y + 5 \;\wedge\; z < y + 2$$

Monadic second-order logic

- First-order variables: interpreted on positions
- Monadic second-order variables: interpreted on sets of positions.
  - Diadic second-order variables: interpreted on relations over positions
  - Monadic third-order variables: interpreted on sets of sets of positions
  - New atomic formulas: $x \in X$

Expressing „even length“

- Express

  There is a set X of positions such that

  – X contains exactly the even positions, and

  – the last position belongs to X.

- Express

  X contains exactly the even positions

  as

A position is in X iff it is second position or the second successor of another position of X

Syntax and semantics of MSO

- New set $\{X, Y, Z, \dots\}$ of second-order variables
- New syntax: $x \in X$ and $\exists x\ \varphi$
- New semantics:
  - Interpretations now also assign sets of positions to the free second-order variables.
  - Satisfaction defined as expected.

Expressing $c^*(ab)^*d^*$

- Express:

There is a block X of consecutive positions such that

  - **before X there are only c's;**
  - **after X there are only b's;**
  - **a's and b's alternate in X;**
  - **the first letter in X is an a, and the last is a b.**

- Then we can take the formula

$$\exists X\ (Cons(X)\ \wedge Boc(X) \wedge Aod(X) \wedge \mathrm{Alt}(X)$$
$$\wedge\, Fa(X) \wedge Lb(X)\ )$$

- X is a block of consecutive positions
- Before X there are only c's
- In X a's and b's alternate

Every regular language is expressible in MSO logic

- Goal: given an arbitrary regular language L, construct an MSO sentence φ such having $L = L(\varphi)$.
- We use: if L is regular, then there is a DFA A recognizing L.
- Idea: construct a formula expressing

  the run of A on this word is accepting

- Fix a regular language L.
- Fix a DFA A with states $q_0, \ldots, q_n$ recognizing L.
- Fix a word $w = a_1 a_2 \ldots a_m$.
- Let P_q be the set of positions i such that after reading $a_1 a_2 \ldots a_i$ the automaton A is in state q.
- We have:

  A accepts w iff $m \in P_q$ for some final state q.