PART 9: SEMANTIC AUTOMATA

Sections 9.1 – 9.5 are background from the class notes Quantification and Modality.

9.1. GENERALIZED QUANTIFIERS

DET = {EVERY, SOME, NO, n, AT MOST n, AT LEAST n, EXACTLY n,
BETWEEN n AND m, MOST} where $n,m \in N$ and $m > n$.

ABSTRACTION:

If $x \in VAR$ and $\varphi \in FORM$, then $\lambda x\varphi \in PRED^1$

QUANTIFICATION:

If $D \in DET$ and $P,Q \in PRED^1$, then $D(P,Q) \in FORM$

For model $M = <D_M,F_M>$ and assignment function g:

For every $D \in DET$: $[\![D]\!]_{M,g} = F_M(D)$

If $x \in VAR$ and $\varphi \in FORM$, then:

$[\![\lambda x\varphi]\!]_{M,g} = \{d \in D_M: [\![\varphi]\!]_{M,gx}{}^{d} = 1\}$

If $D \in DET$ and $P,Q \in PRED^1$, then:

$[\![D(P,Q)]\!]_{M,g} = 1$ iff $< [\![P]\!]_{M,g}, [\![Q]\!]_{M,g} > \in [\![D]\!]_{M,g}$

This leaves the specification of the lexical items, the determiners:

For every $D \in DET$: $F_M(D) \subseteq pow(D_M) \times pow(D_M)$

Every determiner is interpreted as a **relation** between **sets** of individuals.

F_M(EVERY) $= \{<X,Y>: X,Y \subseteq D_M$ and $\mathbf{X \subseteq Y}\}$

F_M(SOME) $= \{<X,Y>: X,Y \subseteq D_M$ and $\mathbf{X \cap Y \neq Ø}\}$

F_M(NO) $= \{<X,Y>: X,Y \subseteq D_M$ and $\mathbf{X \cap Y = Ø}\}$

F_M(AT LEAST n) $= \{<X,Y>: X,Y \subseteq D_M$ and $\mathbf{|X \cap Y| \geq n}\}$

nF_M(AT MOST n) $= \{<X,Y>: X,Y \subseteq D_M$ and $\mathbf{|X \cap Y| \leq n}\}$

F_M(n) $= F_M$(AT LEAST n)

F_M(EXACTLY n) $= \{<X,Y>: X,Y \subseteq D_M$ and $\mathbf{|X \cap Y| = n}\}$

F_M(BETWEEN n AND m)$= \{<X,Y>: X,Y \subseteq D_M$ and $\mathbf{n \leq |X \cap Y| \leq m}\}$

F_M(MOST) $= \{<X,Y>: X,Y \subseteq D_M$ and $\mathbf{|X \cap Y| > |X - Y|}\}$

9.2. GENERAL CONSTRAINTS ON DETERMINER INTERPRETATION.

With some notorious problematic cases, discussed in the literature (eg. *few, many*), natural language determiners all satisfy the following principles of **extension**, **conservativity** and **quantity** (van Benthem 1983).

EXTENSION

> Determiner α satisfies **extension** iff for all models M_1, M_2 and
> for all sets X,Y such that $X,Y \subseteq D_{M1}$ and $X,Y \subseteq D_{M2}$:
> $<X,Y> \in F_{M1}(\alpha)$ iff $<X,Y> \in F_{M2}(\alpha)$

Let $F_{M1}(P) = F_{M2}(P) = X$ and $F_{M1}(Q) = F_{M2}(Q) = Y$.
If α satisfies extension, then the truthvalue of $\alpha(P,Q)$ depends only on what is in $X \cup Y$, not on what is in $D_{M1} - (X \cup Y)$ or in $D_{M2} - (X \cup Y)$.

The intuition is the following:
If α satisfies extension then, if we **only** specify of a model $F_M(BOY)$ and $F_M(SING)$, the truth value of $\alpha(BOY,SING)$ in M is already determined.

This is a natural constraint on natural language determiners:
The truth value of *every boy/some boy/no boy/most boys…sing(s)* does **not** depend on the presence or absence of objects that are neither boys nor singers.

CONSERVATIVITY

> Determiner α is **conservative** iff for every model M and
> for all sets $X,Y \subseteq D_M$:
> $<X,Y> \in F_M(\alpha)$ iff $<X,X \cap Y> \in F_M(\alpha)$

There is another formulation of conservativity and extension, which is useful:

> Determiner α satisfies **extension** and **conservativity** iff
> for all models M_1,M_2, and all sets X_1,Y_1,X_2,Y_2 such that
> $X_1, Y_1 \subseteq D_{M1}$ and $X_2, Y_2 \subseteq D_{M2}$:
> If $X_1 \cap Y_1 = X_2 \cap Y_2$ and $X_1 - Y_1 = X_2 - Y_2$ then
> $<X_1,Y_1> \in F_{M1}(\alpha)$ iff $<X_2,Y_2> \in F_{M2}(\alpha)$.

Let $F_{M1}(P) = X_1$ and $F_{M2}(P) = Y_1$ and $F_{M1}(Q) = X_2$ and$F_{M2}(Q) = Y_2$.
If α satisfies extension, and conservativity, then the truthvalue of $\alpha(P,Q)$ depends only on what is in $X_1 \cap Y_1$ (= $X_2 \cap Y_2$) and in $X_1 - Y_1$ (= $X_2 - Y_2$).

The intuition is the following:
If α satisfies extension and conservativity, then if we specify of a model M, **not even** what $F_M(BOY)$ and $F_M(SING)$ are, **but only** what $F_M(BOY) \cap F_M(SING)$ and $F_M(BOY) - F_M(SING)$ are, then **still** the truth value of $\alpha(BOY,SING)$ in M is already determined.

This is a natural constraint on natural language determiners:

The truth value of *every boy/some boy/no boy/most boys…sing(s)* does **not** depend on the presence or absence of objects that are neither boys nor singers, **and also not** on the presence or absence of singers that are not boys: it only depends on what is in the set of boys that sing, and what is in the set of boys that don't sing.

Conservativity can be checked in the following pattern:

> α is conservative iff $\alpha(BOY,WALK)$ is equivalent to
> $\alpha(BOY,\lambda x BOY(x) \wedge WALK(x))$
>
> α boy walks iff α boy is a boy that walks
> α boys walk iff α boys are boys that walk.

cf:

> Every boy walks iff Every boy is a boy that walks
> Most boys walk iff Most boys are boys that walk.

QUANTITY (Independent definition in terms of permutations, see van Benthem).

> Determiner α satisfies **extension** and **conservativity** and **quantity** iff
> for all models M_1,M_2, and all sets X_1,Y_1,X_2,Y_2 such that
> $X_1, Y_1 \subseteq D_{M1}$ and $X_2, Y_2 \subseteq D_{M2}$:
> If $|X_1 \cap Y_1| = |X_2 \cap Y_2|$ and $|X_1 - Y_1| = |X_2 - Y_2|$ then
> $<X_1,Y_1> \in F_{M1}(\alpha)$ iff $<X_2,Y_2> \in F_{M2}(\alpha)$.

Let $F_{M1}(P) = X_1$ and $F_{M2}(P) = Y_1$ and $F_{M1}(Q) = X_2$ and$F_{M2}(Q) = Y_2$.
If α satisfies extension, and conservativity and extension , then the truthvalue of $\alpha(P,Q)$ depends only on **the cardinality of** $X_1 \cap Y_1$ (= $|X_2 \cap Y_2|$) and **the cardinality of** $X_1 - Y_1$ (= $|X_2 - Y_2|$).

The intuition is the following:
If α satisfies extension and conservativity and quantity, then if we specify of a model M, **not even** what $F_M(BOY)$ and $F_M(SING)$ are, **and not even** what $F_M(BOY) \cap F_M(SING)$ and $F_M(BOY) - F_M(SING)$ are, **but only** what $|F_M(BOY) \cap F_M(SING)|$ and $|F_M(BOY) - F_M(SING)|$ are
then **still** the truth value of $\alpha(BOY,SING)$ in M is already determined.

This is a natural constraint on natural language determiners:
The truth value of *every boy/some boy/no boy/most boys…sing(s)* does **not** depend on the presence or absence of objects that are neither boys nor singers, **and also not** on the presence or absence of singers that are not boys; **it doesn't even** depend on **what** is in the set of boys that sing, and **what** is in the set of boys that don't sing, **but only** on **how many** things there are in the set of boys that sing and on **how many** things there are in the set of boys that don't sing.

For determiners that satisfy extension, conservativity and quantity we can set up the semantics in the following more general way.

We let the model M associate with every determiner α that satisfies extension, conservativity and quantity a **relation r_α between numbers**, $r_\alpha \subseteq N \times N$.
We associate for every model the same relation r_α with α.
In terms of this, we **define** $F_M(\alpha)$:

$$F_M(\alpha) = \{ \langle X,Y\rangle : X,Y \in D_M \text{ and } \langle |X\cap Y|,|X-Y|\rangle \in r_\alpha \}$$

Given this, the meaning of the determiner α is now reduced to the relation r_α between numbers. These meanings we specify as follows:

r_{EVERY} = $\{\langle n,0\rangle : n \in N\}$
r_{SOME} = $\{\langle n,m\rangle : n,m \in N \text{ and } n>0\}$
r_{NO} = $\{\langle 0,m\rangle : m \in N\}$
$r_{AT\ LEAST\ k}$ = $\{\langle n,m\rangle : n,m \in N \text{ and } n\geq k\}$ for $k \in N$
$r_{AT\ MOST\ k:}$ = $\{\langle n,m\rangle : n,m \in N \text{ and } n\leq k\}$ for $k \in N$
$r_{EXACTLY\ k:}$ = $\{\langle k,m\rangle : m \in N\}$ for $k \in N$
$r_{BETWEEN\ k\ AND\ p}$= $\{\langle q,m\rangle : q,m \in N \text{ and } k \leq q \leq p\}$ for $k,p \in N$, $k < p$
$r_{MOST\ :}$ = $\{\langle n,m\rangle : n,m \in N \text{ and } n>m\}$

9.3 DETERMINERS AS PATTERNS ON THE TREE OF NUMBERS
(van Benthem 1983)

If |BOY| = 3, then there are four possibilities for the cardinalities in
$\langle|BOY \cap SING|, |BOY - SING|\rangle$:
<0,3> means: $|BOY \cap SING| = 0$ and $|BOY - SING| = 3$
<1,2> means: $|BOY \cap SING| = 1$ and $|BOY - SING| = 2$
<2,1> means: $|BOY \cap SING| = 2$ and $|BOY - SING| = 1$
<3,0> means: $|BOY \cap SING| = 3$ and $|BOY - SING| = 0$

We can write down a **tree of numbers** which shows for each cardinality of BOY, all the possibilities for the cardinalities of $\langle|BOY \cap SING|, |BOY - SING|\rangle$:

<0,0> |BOY|=0
<0,1> <1,0> |BOY|=1
<0,2> <1,1> <2,0> |BOY|=2
<0,3> <1,2> <2,1> <3,0> |BOY|=3
<0,4> <1,3> <2,2> <3,1> <4,0> |BOY|=4
<0,5> <1,4> <2,3> <3,2> <4,1> <5,0> |BOY|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0> |BOY|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0> |BOY|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0> |BOY|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,3> <6,3> <7,2> <8,1> <9,0> |BOY|=9
...   ...

We can now study the **pattern** that each determiner meaning r_α makes on the tree of numbers, by **highlighting** (bold italic) the extension of r_α:

r_{EVERY}

<0,0> |BOY|=0
<0,1> ***<1,0>*** |BOY|=1
<0,2> <1,1> ***<2,0>*** |BOY|=2
<0,3> <1,2> <2,1> ***<3,0>*** |BOY|=3
<0,4> <1,3> <2,2> <3,1> ***<4,0>*** |BOY|=4
<0,5> <1,4> <2,3> <3,2> <4,1> ***<5,0>*** |BOY|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> ***<6,0>*** |BOY|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> ***<7,0>*** |BOY|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> ***<8,0>*** |BOY|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,3> <6,3> <7,2> <8,1> ***<9,0>*** |BOY|=9
… …

r_{SOME}

<0,0> |BOY|=0
<0,1> ***<1,0>*** |BOY|=1
<0,2> ***<1,1> <2,0>*** |BOY|=2
<0,3> ***<1,2> <2,1> <3,0>*** |BOY|=3
<0,4> ***<1,3> <2,2> <3,1> <4,0>*** |BOY|=4
<0,5> ***<1,4> <2,3> <3,2> <4,1> <5,0>*** |BOY|=5
<0,6> ***<1,5> <2,4> <3,3> <4,2> <5,1> <6,0>*** |BOY|=6
<0,7> ***<1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0>*** |BOY|=7
<0,8> ***<1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0>*** |BOY|=8
<0,9> ***<1,8> <2,7> <3,6> <4,5> <5,3> <6,3> <7,2> <8,1> <9,0>*** |BOY|=9
…

r_{NO}

<0,0> |BOY|=0
<0,1> <1,0> |BOY|=1
<0,2> <1,1> <2,0> |BOY|=2
<0,3> <1,2> <2,1> <3,0> |BOY|=3
<0,4> <1,3> <2,2> <3,1> <4,0> |BOY|=4
<0,5> <1,4> <2,3> <3,2> <4,1> <5,0> |BOY|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0> |BOY|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0> |BOY|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0> |BOY|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,3> <6,3> <7,2> <8,1> <9,0> |BOY|=9
… …

$r_{\text{AT LEAST 4}}$

<0,0> |BOY|=0
<0,1> <1,0> |BOY|=1
<0,2> <1,1> <2,0> |BOY|=2
<0,3> <1,2> <2,1> <3,0> |BOY|=3
<0,4> <1,3> <2,2> <3,1> ***<4,0>*** |BOY|=4
<0,5> <1,4> <2,3> <3,2> ***<4,1> <5,0>*** |BOY|=5
<0,6> <1,5> <2,4> <3,3> ***<4,2> <5,1> <6,0>*** |BOY|=6
<0,7> <1,6> <2,5> <3,4> ***<4,3> <5,2> <6,1> <7,0>*** |BOY|=7
<0,8> <1,7> <2,6> <3,5> ***<4,4> <5,3> <6,2> <7,1> <8,0>*** |BOY|=8
<0,9> <1,8> <2,7> <3,6> ***<4,5> <5,3> <6,3> <7,2> <8,1> <9,0>*** |BOY|=9
… …

$r_{\text{AT MOST 4}}$

<0,0> |BOY|=0
<0,1> <1,0> |BOY|=1
<0,2> <1,1> <2,0> |BOY|=2
<0,3> <1,2> <2,1> <3,0> |BOY|=3
<0,4> <1,3> <2,2> <3,1> <4,0> |BOY|=4
<0,5> <1,4> <2,3> <3,2> <4,1> <5,0> |BOY|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0> |BOY|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0> |BOY|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0> |BOY|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,3> <6,3> <7,2> <8,1> <9,0> |BOY|=9
… …

$r_{\text{EXACTLY 4}}$

<0,0> |BOY|=0
<0,1> <1,0> |BOY|=1
<0,2> <1,1> <2,0> |BOY|=2
<0,3> <1,2> <2,1> <3,0> |BOY|=3
<0,4> <1,3> <2,2> <3,1> ***<4,0>*** |BOY|=4
<0,5> <1,4> <2,3> <3,2> ***<4,1>*** <5,0> |BOY|=5
<0,6> <1,5> <2,4> <3,3> ***<4,2>*** <5,1> <6,0> |BOY|=6
<0,7> <1,6> <2,5> <3,4> ***<4,3>*** <5,2> <6,1> <7,0> |BOY|=7
<0,8> <1,7> <2,6> <3,5> ***<4,4>*** <5,3> <6,2> <7,1> <8,0> |BOY|=8
<0,9> <1,8> <2,7> <3,6> ***<4,5>*** <5,3> <6,3> <7,2> <8,1> <9,0> |BOY|=9
… …

r_{BETWEEN 2 and 4}

<0,0>	|BOY|=0
<0,1> <1,0>	|BOY|=1
<0,2> <1,1> ***<2,0>***	|BOY|=2
<0,3> <1,2> ***<2,1> <3,0>***	|BOY|=3
<0,4> <1,3> ***<2,2> <3,1> <4,0>***	|BOY|=4
<0,5> <1,4> ***<2,3> <3,2> <4,1>*** <5,0>	|BOY|=5
<0,6> <1,5> ***<2,4> <3,3> <4,2>*** <5,1> <6,0>	|BOY|=6
<0,7> <1,6> ***<2,5> <3,4> <4,3>*** <5,2> <6,1> <7,0>	|BOY|=7
<0,8> <1,7> ***<2,6> <3,5> <4,4>*** <5,3> <6,2> <7,1> <8,0>	|BOY|=8
<0,9> <1,8> ***<2,7> <3,6> <4,5>*** <5,3> <6,3> <7,2> <8,1> <9,0>	|BOY|=9
...	...

r_{MOST}

<0,0>	|BOY|=0
<0,1> ***<1,0>***	|BOY|=1
<0,2> <1,1> ***<2,0>***	|BOY|=2
<0,3> <1,2> ***<2,1> <3,0>***	|BOY|=3
<0,4> <1,3> <2,2> ***<3,1> <4,0>***	|BOY|=4
<0,5> <1,4> <2,3> ***<3,2> <4,1> <5,0>***	|BOY|=5
<0,6> <1,5> <2,4> <3,3> ***<4,2> <5,1> <6,0>***	|BOY|=6
<0,7> <1,6> <2,5> <3,4> ***<4,3> <5,2> <6,1> <7,0>***	|BOY|=7
<0,8> <1,7> <2,6> <3,5> <4,4> ***<5,3> <6,2> <7,1> <8,0>***	|BOY|=8
<0,9> <1,8> <2,7> <3,6> <4,5> ***<5,3> <6,3> <7,2> <8,1> <9,0>***	|BOY|=9
...	...

9.3. SYMMETRY AND MONOTONICITY

SYMMETRY

Determiner α is **symmetric** iff for every model M and all sets $X,Y \in D_M$:
$<X,Y> \in F_M(\alpha)$ iff $<Y,X> \in F_M(\alpha)$

Pattern: α(BOY,SING) is equivalent to α(SING,BOY)

α boy sings iff α singer is a boy
α boys sing iff α singers are boys

Technically: $F_M(\alpha)$ only depends on $|A \cap B|$: Symmetry follows from commutativity of $\cap$.

	SYMMETRIC
every	NO
some	YES
no	YES
at least n	YES
at most n	YES
exactly n	YES
most	NO

Let us define: $[\![\text{Exist}]\!]_{M,g} = D_M$

Then we have the following for symetric DETS:

$\text{DET}(A,B) \Leftrightarrow_{\text{conservativity}} \text{DET}(A,A\cap B) \Leftrightarrow_{\text{symmetry}} \text{DET}(A\cap B,A)$
$\Leftrightarrow_{\text{conservativity}} \text{DET}(A\cap B,A\cap B) \Leftrightarrow <|(A\cap B) \cap (A\cap B)|, |(A\cap B) - (A\cap B)|> \in r_{\text{DET}}$
$\Leftrightarrow <|A\cap B|,0> \in r_{\text{DET}}$

$\text{DET}(A\cap B,\text{EXIST}) \Leftrightarrow <|(A\cap B) \cap D_M|,|(A\cap B) - D_M| \in r_{\text{DET}}$
$\Leftrightarrow <|A\cap B|,0> \in r_{\text{DET}}$

Thus:

D is **symmetric** iff $\text{DET}(A,B) \Leftrightarrow \text{DET}(A\cap B,\text{EXIST})$

This means that the truth conditions of DET(A,B) **only** depend on the cardinality of $A\cap B$, ie. are completely determined by that.

MONOTONICITY.
Let α be a determiner.
In $\alpha(P,Q)$ we call P the **first argument** of α and Q the **second argument** of α

Terminology:
α is $\uparrow_1$: α is **upward monotonic**, **upward entailing, on its first argument**
α is $\downarrow_1$: α is **downward monotonic**, **downward entailing, on its first argument**
α is $-_1$: α is **neither upward nor downward monotonic on its first argument**

α is $\uparrow_2$: α is **upward monotonic**, **upward entailing, on its second argument**
α is $\downarrow_2$: α is **downward monotonic**, **downward entailing, on its second argument**
α is $-_2$ α is **neither upward nor downward monotonic on its second argument**

α is $\boldsymbol{\uparrow_1}$ iff for every model M and all sets $X_1,X_2,Y \subseteq D_M$:
if $<\mathbf{X_1},Y> \in F_M(\alpha)$ and $\mathbf{X_1 \subseteq X_2}$ then $<\mathbf{X_2},Y> \in F_M(\alpha)$

α is $\boldsymbol{\downarrow_1}$ iff for every model M and all sets $X_1,X_2,Y \subseteq D_M$:
if $<\mathbf{X_2},Y> \in F_M(\alpha)$ and $\mathbf{X_1 \subseteq X_2}$ then $<\mathbf{X_1},Y> \in F_M(\alpha)$

α is $\boldsymbol{-_1}$ iff α is not $\uparrow_1$ and α is not $\downarrow_1$

α is $\boldsymbol{\uparrow_2}$ iff for every model M and all sets $X,Y_1,Y_2 \subseteq D_M$:
if $<X,\mathbf{Y_1}> \in F_M(\alpha)$ and $\mathbf{Y_1 \subseteq Y_2}$ then $<X,\mathbf{Y_2}> \in F_M(\alpha)$

α is $\boldsymbol{\downarrow_2}$ iff for every model M and all sets $X,Y_1,Y_2 \subseteq D_M$:
if $<X,\mathbf{Y_2}> \in F_M(\alpha)$ and $\mathbf{Y_1 \subseteq Y_2}$ then $<X,\mathbf{Y_1}> \in F_M(\alpha)$

α is $\boldsymbol{-_2}$ iff α is not $\uparrow_2$ and α is not $\downarrow_2$

	ARGUMENT 1	ARGUMENT 2
every	↓	↑
some	↑	↑
no	↓	↓
at least n	↑	↑
at most n	↓	↓
exactly n	–	–
most	–	↑

9.4. SYMMETRY AS A PATTERN ON THE TREE OF NUMBERS

α is symmetric iff r_α is symmetric.

r_α is symmetric iff for every n,m≥0: $\langle n,m\rangle \in r_\alpha$ iff $\langle n,0\rangle \in r_\alpha$

i.e.
FACT: if α satisfies EXT, CONS, QUANT, then
α is symmetric iff for every M for every X,Y: whether $\langle X,Y\rangle$ is in $F_M(\alpha)$ or not depends **only** on $|X \cap Y|$.

In terms of the tree of numbers this means that:

r_α is symmetric iff for every n: **either** for every m: $\langle n,m\rangle \in r_\alpha$
or for every m: $\langle n,m\rangle \notin r_\alpha$

In terms of the tree of numbers this means the following.
For number n, $\{\langle n,k\rangle : k \in N\}$ is a **diagonal line** in the tree going from left below to right up:
Like, for n = 3:

<0,0> |BOY|=0

<0,1> <1,0> |BOY|=1

<0,2> <1,1> <2,0> |BOY|=2

<0,3> <1,2> <2,1> <3,0> |BOY|=3

<0,4> <1,3> <2,2> <3,1> <4,0> |BOY|=4

<0,5> <1,4> <2,3> <3,2> <4,1> <5,0> |BOY|=5

<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0> |BOY|=6

<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0> |BOY|=7

<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0> |BOY|=8

<0,9> <1,8> <2,7> <3,6> <4,5> <5,3> <6,3> <7,2> <8,1> <9,0> |BOY|=9

... ...

r_α is symmetric iff every such diagonal line is **either completely inside r_α or completely outside r_α.**

With this we can check straighforwardly in the trees which r_α's are symmetric:

r_{every} is **not** symmetric:

r_{EVERY}

<0,0>	|BOY|=0
<0,1> ***<1,0>***	|BOY|=1
<0,2> <1,1> ***<2,0>***	|BOY|=2
<0,3> <1,2> <2,1> ***<3,0>***	|BOY|=3
<0,4> <1,3> <2,2> <3,1> ***<4,0>***	|BOY|=4
<0,5> <1,4> <2,3> <3,2> <4,1> ***<5,0>***	|BOY|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> ***<6,0>***	|BOY|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> ***<7,0>***	|BOY|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> ***<8,0>***	|BOY|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,3> <6,3> <7,2> <8,1> ***<9,0>***	|BOY|=9
...	...

r_{some} **is** symmetric:

r_{SOME}

<0,0>	|BOY|=0
<0,1> ***<1,0>***	|BOY|=1
<0,2> ***<1,1>*** ***<2,0>***	|BOY|=2
<0,3> ***<1,2>*** ***<2,1>*** ***<3,0>***	|BOY|=3
<0,4> ***<1,3>*** ***<2,2>*** ***<3,1>*** ***<4,0>***	|BOY|=4
<0,5> ***<1,4>*** ***<2,3>*** ***<3,2>*** ***<4,1>*** ***<5,0>***	|BOY|=5
<0,6> ***<1,5>*** ***<2,4>*** ***<3,3>*** ***<4,2>*** ***<5,1>*** ***<6,0>***	|BOY|=6
<0,7> ***<1,6>*** ***<2,5>*** ***<3,4>*** ***<4,3>*** ***<5,2>*** ***<6,1>*** ***<7,0>***	|BOY|=7
<0,8> ***<1,7>*** ***<2,6>*** ***<3,5>*** ***<4,4>*** ***<5,3>*** ***<6,2>*** ***<7,1>*** ***<8,0>***	|BOY|=8
<0,9> ***<1,8>*** ***<2,7>*** ***<3,6>*** ***<4,5>*** ***<5,3>*** ***<6,3>*** ***<7,2>*** ***<8,1>*** ***<9,0>***	|BOY|=9
...	

r_{no} **is** symmetric:

r_{NO}

<0,0>	|BOY|=0
<0,1> <1,0>	|BOY|=1
<0,2> <1,1> <2,0>	|BOY|=2
<0,3> <1,2> <2,1> <3,0>	|BOY|=3
<0,4> <1,3> <2,2> <3,1> <4,0>	|BOY|=4
<0,5> <1,4> <2,3> <3,2> <4,1> <5,0>	|BOY|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0>	|BOY|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0>	|BOY|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0>	|BOY|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,3> <6,3> <7,2> <8,1> <9,0>	|BOY|=9
...	...

It is easy to check that $r_{\leq n}$, $r_{\geq n}$, $r_{=n}$ are symmetric, but that r_{most} is not symmetric.

9.5. MONOTONICITY PATTERNS ON THE TREE OF NUMBERS

$\mathbf{r_\alpha}$ **is** $\uparrow_2$ iff if $<n,m> \in r_\alpha$ then $<n+1,m-1> \in r_\alpha$ and if $n+m = p+q$ and $p \geq n$ and $q \leq m$ then $<p,q> \in r_\alpha$

This means that $\mathbf{r_\alpha}$ **is** $\uparrow_2$ iff if $<n,m> \in r_\alpha$ then any point to the **right on that same line** is also in r_α

Example: $r_{\geq 4}$ is $\uparrow_2$:

$\mathbf{r_{AT\ LEAST\ 4}}$

<0,0>	|BOY|=0
<0,1> <1,0>	|BOY|=1
<0,2> <1,1> <2,0>	|BOY|=2
<0,3> <1,2> <2,1> <3,0>	|BOY|=3
<0,4> <1,3> <2,2> <3,1> ***<4,0>***	|BOY|=4
<0,5> <1,4> <2,3> <3,2> ***<4,1> <5,0>***	|BOY|=5
<0,6> <1,5> <2,4> <3,3> ***<4,2> <5,1> <6,0>***	|BOY|=6
<0,7> <1,6> <2,5> <3,4> ***<4,3> <5,2> <6,1> <7,0>***	|BOY|=7
<0,8> <1,7> <2,6> <3,5> ***<4,4> <5,3> <6,2> <7,1> <8,0>***	|BOY|=8
<0,9> <1,8> <2,7> <3,6> ***<4,5> <5,3> <6,3> <7,2> <8,1> <9,0>***	|BOY|=9
...	...

$\mathbf{r_\alpha}$ **is** $\downarrow_2$ iff if $<n,m> \in r_\alpha$ then and $<n-1,m+1> \in r_\alpha$ and if $n+m = p+q$ and $p \leq n$ and $q \geq m$ then $<p,q> \in r_\alpha$

This means that $\mathbf{r_\alpha}$ **is** $\downarrow_2$ iff if $<n,m> \in r_\alpha$ then any point to the **left on that same line** is also in r_α

Example: $r_{\leq 4}$ is $\downarrow_2$:

$\mathbf{r_{AT\ MOST\ 4}}$

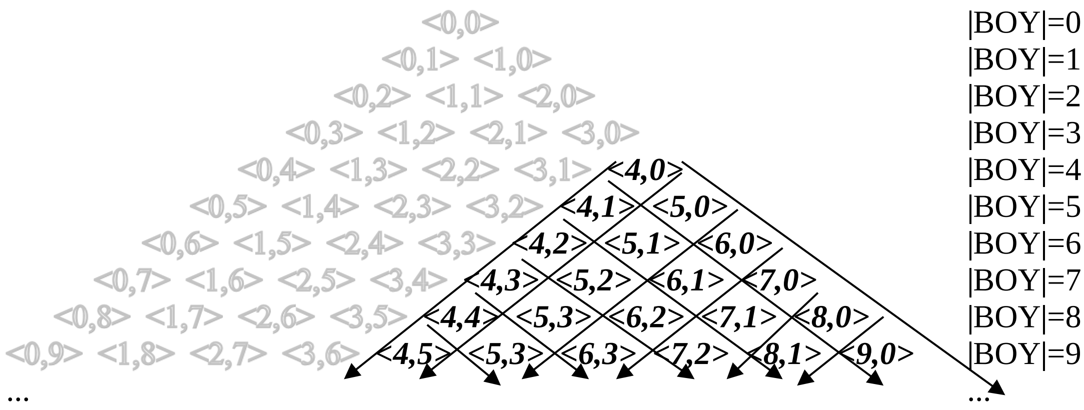
<0,0>	|BOY|=0
<0,1> <1,0>	|BOY|=1
<0,2> <1,1> <2,0>	|BOY|=2
<0,3> <1,2> <2,1> <3,0>	|BOY|=3
<0,4> <1,3> <2,2> <3,1> <4,0>	|BOY|=4
<0,5> <1,4> <2,3> <3,2> <4,1> <5,0>	|BOY|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0>	|BOY|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0>	|BOY|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0>	|BOY|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,3> <6,3> <7,2> <8,1> <9,0>	|BOY|=9
...	...

$\mathbf{r}_{\alpha}$ **is** $\boldsymbol{\uparrow}_{\mathbf{1}}$ iff if $\langle n,m\rangle \in r_{\alpha}$ then $\langle n+1,m\rangle$, $\langle n,m+1\rangle \in r_{\alpha}$

This means that $\mathbf{r}_{\alpha}$ **is** $\boldsymbol{\uparrow}_{\mathbf{1}}$ iff if $\langle n,m\rangle \in r_{\alpha}$ then **the whole triangle with top** $\boldsymbol{\langle n,m\rangle}$ is in r_{α}.

Example: $r_{\geq 4}$ is $\uparrow_1$:

$\mathbf{r}_{\textbf{AT LEAST 4}}$

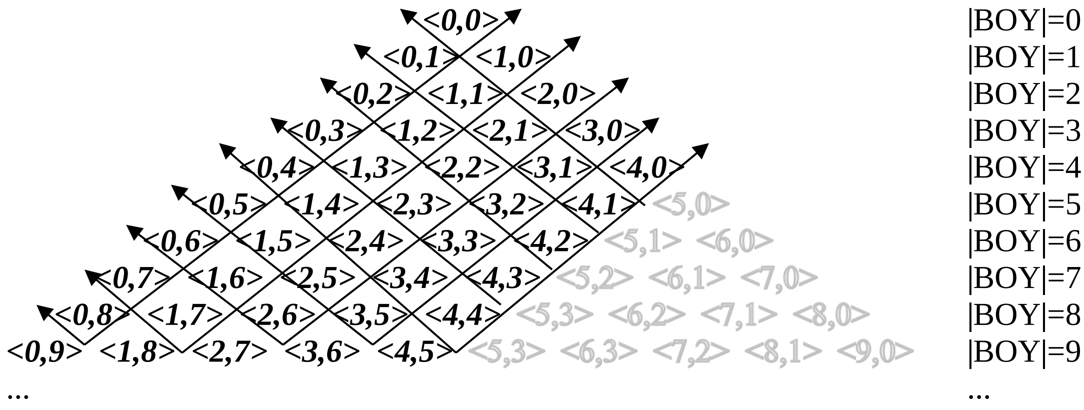


$\mathbf{r}_{\alpha}$ **is** $\boldsymbol{\downarrow}_{\mathbf{1}}$ iff if $\langle n,m\rangle \in r_{\alpha}$ then $\langle n-1,m\rangle$, $\langle n,m-1\rangle \in r_{\alpha}$
(when n or m is 0, set n−1, m−1 to 0 as well)

This means that $\mathbf{r}_{\alpha}$ **is** $\boldsymbol{\downarrow}_{\mathbf{1}}$ iff if $\langle n,m\rangle \in r_{\alpha}$ then **the whole inverted triangle with bottom** $\boldsymbol{\langle n,m\rangle}$ is in r_{α}.

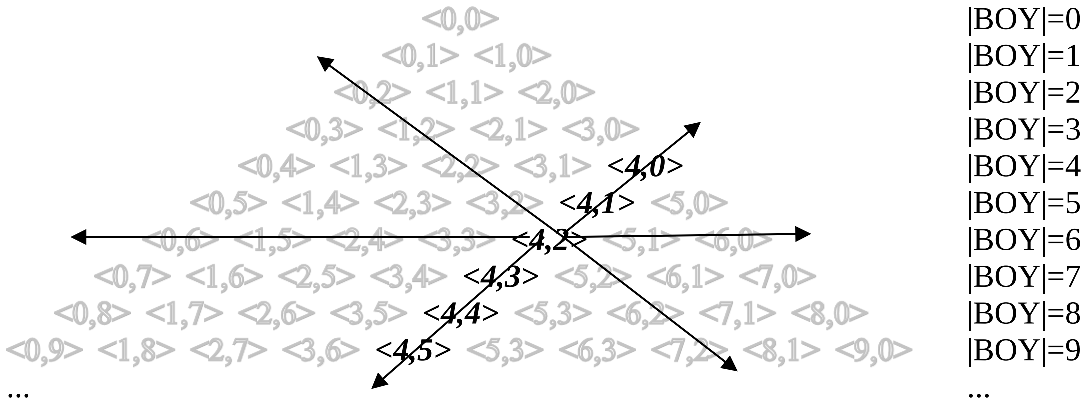
Example: $r_{\leq 4}$ is $\downarrow_1$:

$\mathbf{r}_{\textbf{AT MOST 4}}$



It is easy to check that $r_{=3}$ is none of the above:

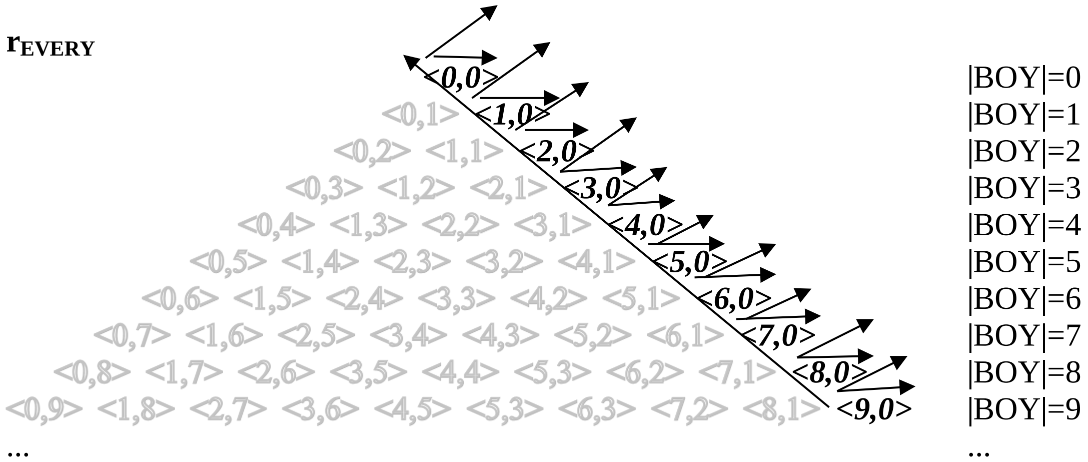
r$_{\textbf{EXACTLY 4}}$



The same for *between 2 and 4*.

r_{every} is $\uparrow_2$, because trivially every point to the right is in (since there are no points to the right).
r_{every} is clearly not $\uparrow_1$, since the downward triangles are not preserved.
r_{every} is $\downarrow_1$, since the upward inverted triangle is just the right edge.

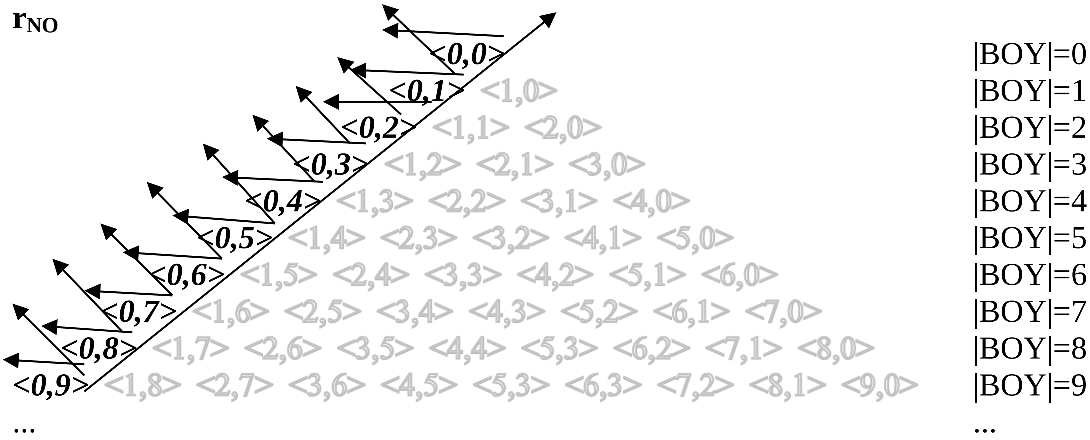
r$_{\textbf{EVERY}}$



r_{no} is $\downarrow_2$ because, again, trivially every point to the left is in.

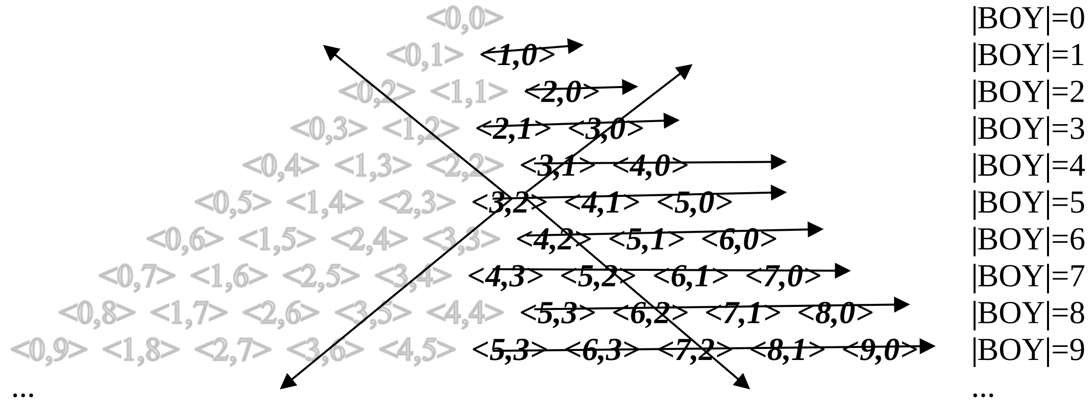
r_{no} is again clearly not $\uparrow_1$, but it is $\downarrow_1$, because, again, the upward inverted triangle is just the left edge.

r$_{NO}$



r_{most} is $\uparrow_2$, but neither $\uparrow_1$ nor $\downarrow_1$: for no point in r_{most} is the downward triangle completely in r_{most} and for no point is the upward triangle completely in r_{most} (because <0,0> is not).

r$_{MOST}$



9.6 SEMANTIC AUTOMATA (van Benthem 1987)

Let α be a string in alphabet Σ, L a language in Σ.

β is a **permutation of** α iff $\psi_\Sigma(\beta)=\psi_\Sigma(\alpha)$

A permutation of α is a string with the same length characteristics: the same number of each symbol, but possibly in a different order.

$\pi(\alpha) = \{\beta \in \Sigma^*: \psi_\Sigma(\beta)=\psi_\Sigma(\alpha)\}$
The **permutation closure** of string α is the set of all permutations of α.
(more properly, the permutation closure of $\{\alpha\}$.)

$\pi(L) = \cup\{\pi(\alpha): \alpha \in L\}$
The **permutation closure** of language L is the union of the permutation closures of the strings in L.

L is **permutation closed** iff $L = \pi(L)$

We will call a permutation closed language a π-language.

We will be interested in permutation closed languages in alphabet {0,1}, i.e. permutation closed subsets of {0,1}.
We will call these languages π-{0,1} languages.

Look at the tree of numbers:

<0,0>	$\lvert BOY\rvert=0$
<0,1> <1,0>	$\lvert BOY\rvert=1$
<0,2> <1,1> <2,0>	$\lvert BOY\rvert=2$
<0,3> <1,2> <2,1> <3,0>	$\lvert BOY\rvert=3$
<0,4> <1,3> <2,2> <3,1> <4,0>	$\lvert BOY\rvert=4$
<0,5> <1,4> <2,3> <3,2> <4,1> <5,0>	$\lvert BOY\rvert=5$
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0>	$\lvert BOY\rvert=6$
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0>	$\lvert BOY\rvert=7$
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0>	$\lvert BOY\rvert=8$
<0,9> <1,8> <2,7> <3,6> <4,5> <5,3> <6,3> <7,2> <8,1> <9,0>	$\lvert BOY\rvert=9$
...	...

We associate with each pair of numbers a set of strings:

$\ell(\langle n,m\rangle) = \pi(1^n0^m)$

So, for example: $\ell(\langle 2,1\rangle) \rightarrow \pi(110) = \{110, 101, 011\}$
$\ell(\langle 3,2\rangle) \rightarrow \pi(11100) =$
{11100, 11010, 10110, 01110, 11001, 10101, 01101, 10011, 01011, 00111}

Let T be the domain of the tree of numbers. Let $X \subseteq T$:
$\ell(X) = \cup\{\ell(\langle n,m\rangle): \langle n,m\rangle \in X\}$

The intuition is the following: <3,2> indicates that on a domain with five individuals three individuals are in $X \cap Y$ and two individuals are in $X - Y$. On entering $X \cap Y$ an individual is given a T-shirt with a 1, on entering $X - Y$, an individual is given a T-shirt with a 0. In checking the domain, we are not assuming that the objects are given in any preferred order, but, because of quantity, the order doesn't matter: quantifiers are not order-dependent (i.e. we are assuming that *the first five* in *the first five* boys is not a quantifier in the sense we are studying here, nor *every other boy*).

- ℓ associates with each pair in the tree a finite π-{0,1} language.
- ℓ associates with each set of pairs in the tree a π-{0,1} language.

We have associated with quantifiers characteristic patterns on the tree of numbers, highlighting for each quantifier α the extension of relation r_α as a set of pairs of numbers.
With the move to ℓ we can now interpret the extension of relation r_α as a π-{0,1} language.

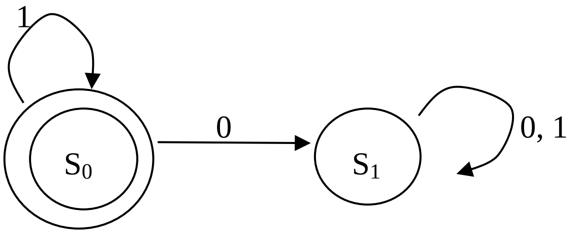
And this means that we can now ask questions like: how complex are the relations r_α corresponding to natural language quantifiers?

[Note: we are not at all entering into the far more complex questions of the complexity of interactions between quantifiers. Here we are interested in the basic quantifiers.]

rEVERY

<0,0>	|BOY|=0
<0,1> ***<1,0>***	|BOY|=1
<0,2> <1,1> ***<2,0>***	|BOY|=2
<0,3> <1,2> <2,1> ***<3,0>***	|BOY|=3
<0,4> <1,3> <2,2> <3,1> ***<4,0>***	|BOY|=4
<0,5> <1,4> <2,3> <3,2> <4,1> ***<5,0>***	|BOY|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> ***<6,0>***	|BOY|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> ***<7,0>***	|BOY|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> ***<8,0>***	|BOY|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,3> <6,3> <7,2> <8,1> ***<9,0>***	|BOY|=9
…	…

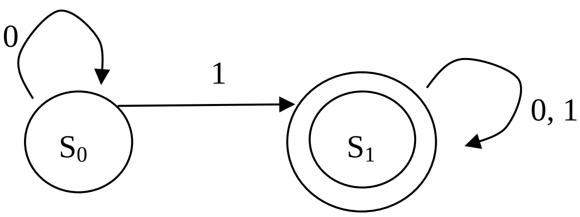
$\ell(\mathbf{r}_{\text{EVERY}}) = \{\alpha \in \{0,1\}^*: |0|_\alpha = 0\} = 1^n, n \geq 0$



rSOME

<0,0>	|BOY|=0
<0,1> ***<1,0>***	|BOY|=1
<0,2> ***<1,1> <2,0>***	|BOY|=2
<0,3> ***<1,2> <2,1> <3,0>***	|BOY|=3
<0,4> ***<1,3> <2,2> <3,1> <4,0>***	|BOY|=4
<0,5> ***<1,4> <2,3> <3,2> <4,1> <5,0>***	|BOY|=5
<0,6> ***<1,5> <2,4> <3,3> <4,2> <5,1> <6,0>***	|BOY|=6
<0,7> ***<1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0>***	|BOY|=7
<0,8> ***<1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0>***	|BOY|=8
<0,9> ***<1,8> <2,7> <3,6> <4,5> <5,3> <6,3> <7,2> <8,1> <9,0>***	|BOY|=9
…	

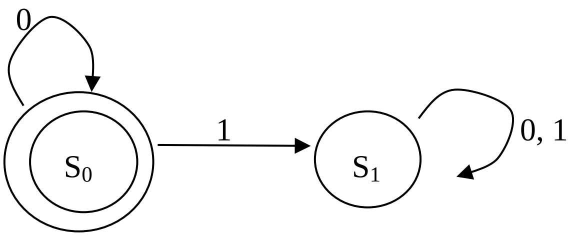
$\ell(\mathbf{r}_{\text{SOME}}) = \{\alpha \in \{0,1\}^*: |1|_\alpha > 0\}$



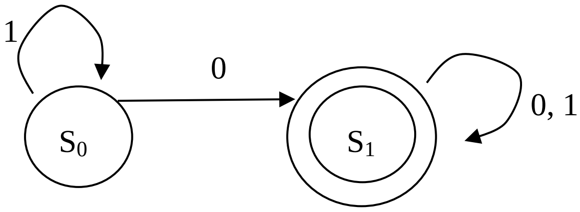
$\mathbf{r}_{NO}$

<0,0>	|BOY|=0
<0,1> <1,0>	|BOY|=1
<0,2> <1,1> <2,0>	|BOY|=2
<0,3> <1,2> <2,1> <3,0>	|BOY|=3
<0,4> <1,3> <2,2> <3,1> <4,0>	|BOY|=4
<0,5> <1,4> <2,3> <3,2> <4,1> <5,0>	|BOY|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0>	|BOY|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0>	|BOY|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0>	|BOY|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,3> <6,3> <7,2> <8,1> <9,0>	|BOY|=9
...	...

$\ell(\mathbf{r}_{NO}) = \{\alpha \in \{0,1\}^*: |1|_\alpha = 0\} = 0^n, n \geq 0$



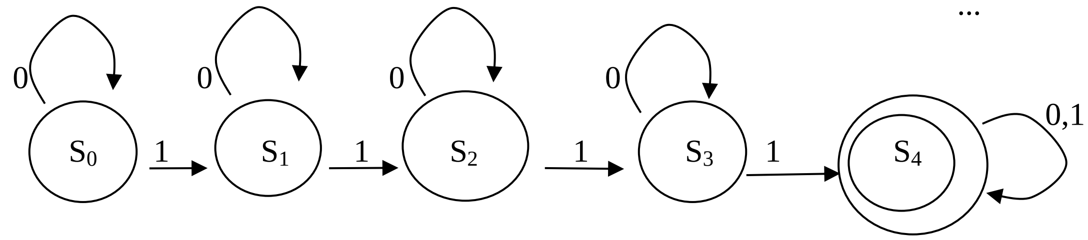
(From this is obvious what the automaton for $\mathbf{r}_{NOT\ EVERY}$ would look like:
$\ell(\mathbf{r}_{NOT\ EVERY}) = \{\alpha \in \{0,1\}^*: |0|_\alpha > 0\}$



The Aristotelian square as a square of automata.

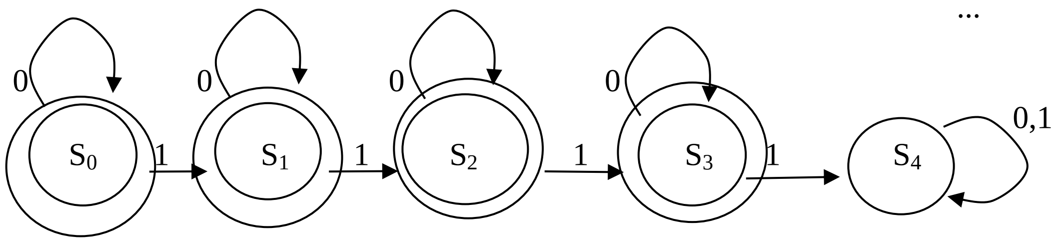
$r_{\text{AT LEAST 4}}$

<0,0>	|BOY|=0
<0,1> <1,0>	|BOY|=1
<0,2> <1,1> <2,0>	|BOY|=2
<0,3> <1,2> <2,1> <3,0>	|BOY|=3
<0,4> <1,3> <2,2> <3,1> ***<4,0>***	|BOY|=4
<0,5> <1,4> <2,3> <3,2> ***<4,1> <5,0>***	|BOY|=5
<0,6> <1,5> <2,4> <3,3> ***<4,2> <5,1> <6,0>***	|BOY|=6
<0,7> <1,6> <2,5> <3,4> ***<4,3> <5,2> <6,1> <7,0>***	|BOY|=7
<0,8> <1,7> <2,6> <3,5> ***<4,4> <5,3> <6,2> <7,1> <8,0>***	|BOY|=8
<0,9> <1,8> <2,7> <3,6> ***<4,5> <5,3> <6,3> <7,2> <8,1> <9,0>***	|BOY|=9
...	...



$r_{\text{AT MOST 4}}$

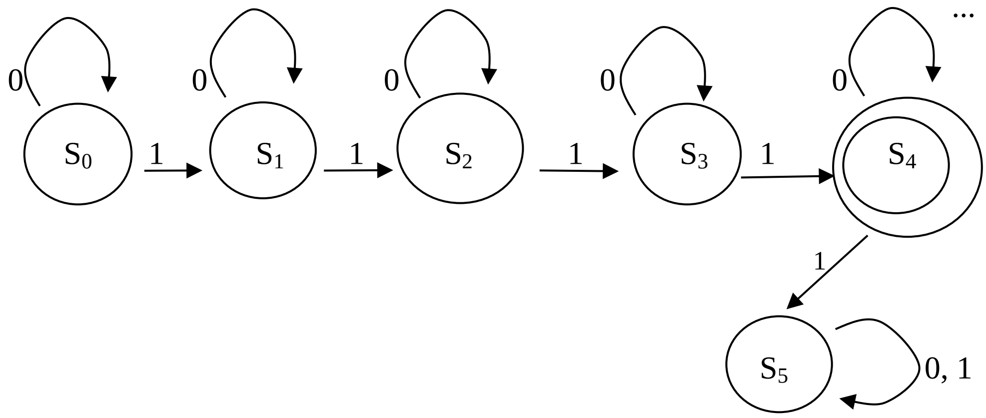
<0,0>	|BOY|=0
<0,1> <1,0>	|BOY|=1
<0,2> <1,1> <2,0>	|BOY|=2
<0,3> <1,2> <2,1> <3,0>	|BOY|=3
<0,4> <1,3> <2,2> <3,1> <4,0>	|BOY|=4
<0,5> <1,4> <2,3> <3,2> <4,1> <5,0>	|BOY|=5
<0,6> <1,5> <2,4> <3,3> <4,2> <5,1> <6,0>	|BOY|=6
<0,7> <1,6> <2,5> <3,4> <4,3> <5,2> <6,1> <7,0>	|BOY|=7
<0,8> <1,7> <2,6> <3,5> <4,4> <5,3> <6,2> <7,1> <8,0>	|BOY|=8
<0,9> <1,8> <2,7> <3,6> <4,5> <5,3> <6,3> <7,2> <8,1> <9,0>	|BOY|=9
...	...
	...



$r_{EXACTLY\ 4}$

<0,0> |BOY|=0
<0,1> <1,0> |BOY|=1
<0,2> <1,1> <2,0> |BOY|=2
<0,3> <1,2> <2,1> <3,0> |BOY|=3
<0,4> <1,3> <2,2> <3,1> ***<4,0>*** |BOY|=4
<0,5> <1,4> <2,3> <3,2> ***<4,1>*** <5,0> |BOY|=5
<0,6> <1,5> <2,4> <3,3> ***<4,2>*** <5,1> <6,0> |BOY|=6
<0,7> <1,6> <2,5> <3,4> ***<4,3>*** <5,2> <6,1> <7,0> |BOY|=7
<0,8> <1,7> <2,6> <3,5> ***<4,4>*** <5,3> <6,2> <7,1> <8,0> |BOY|=8
<0,9> <1,8> <2,7> <3,6> ***<4,5>*** <5,3> <6,3> <7,2> <8,1> <9,0> |BOY|=9
... ...

...



All these quantifiers are first order definable.

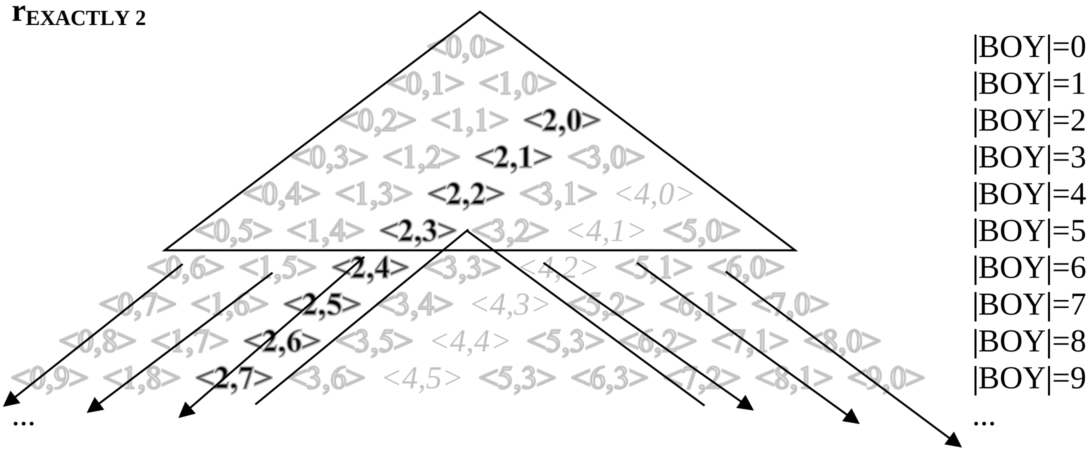
Obviously, we turn this into an automatoin for *between 2 and 4* by making S_2 and S_3 final states as well.

Theorem: All first-order definable quantifiers are recognized by finite state atomata.

Proof sketch: Fraissé proved that all first order definable quantifiers have the insensitivity to domain extension described above: Adding objects to the sets $X \cap Y$ or $X - Y$, may change the truth value of DET(X,Y), but for first order definable quantifiers there is a number such that beyond that, adding more elements to $X \cap Y$ or $X - Y$ doesn't change truth value anymore. In terms of the tree of numbers, the Fraissé property means the following:

I will use the tree for *exactly 2* as our example:
Up to some finite level (the top triangle), the truth values can flip-flop arbitrarily (the top triangle) But at some point (<3,3>) the growth of the sets involved follow the pattern indicated: the truth value of the element <3,3> determines the truth value of the whole triangle it dominates, and to the left and right, the truth values are preserved along the diagonals as indicated.

$r_{\text{EXACTLY 2}}$



Now, the quantifier will accept a subset of the top triangle, which is a finite set, hence regular, plus some of the diagonals, of which there is a finite number, and it is easy to see that each diagonal is a regular set, plus possibly the bottom triangle, which is also a regular set ($\pi(1^n0^m)$) where (in the example) $3 \leq n,m$). Hence the language corresponding to any first-order definable quantifier is regular.

You can check that *most* does not satisfy the Fraissé characterization, and hence *most* is not first-order definable:

r_{MOST}

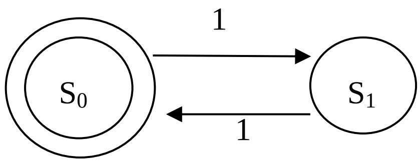
<0,0> |BOY|=0
<0,1> ***<1,0>*** |BOY|=1
<0,2> <1,1> ***<2,0>*** |BOY|=2
<0,3> <1,2> ***<2,1> <3,0>*** |BOY|=3
<0,4> <1,3> <2,2> ***<3,1> <4,0>*** |BOY|=4
<0,5> <1,4> <2,3> ***<3,2> <4,1> <5,0>*** |BOY|=5
<0,6> <1,5> <2,4> <3,3> ***<4,2> <5,1> <6,0>*** |BOY|=6
<0,7> <1,6> <2,5> <3,4> ***<4,3> <5,2> <6,1> <7,0>*** |BOY|=7
<0,8> <1,7> <2,6> <3,5> <4,4> ***<5,3> <6,2> <7,1> <8,0>*** |BOY|=8
<0,9> <1,8> <2,7> <3,6> <4,5> ***<5,3> <6,3> <7,2> <8,1> <9,0>*** |BOY|=9
... ...

$\ell(\mathbf{r}_{MOST}) = \{\alpha \in \{0,1\}^*: |1|_\alpha > |0|_\alpha\}$

We have seen this language before. It is not a regular language. We can prove this with the pumping lemma for regular languages. Assume the language is regular, and take a string 0^k1^{k+1} with k>n. You need to be able to find a division $\alpha\beta\gamma$, with $|\alpha\beta|\leq n$ such that $\alpha\beta^i\gamma$ is in the language. But this means that β can only consist of 0's, and, of course, pumping will take you out of the language. We have given a pushdown storage automaton for this language, so it is context free.

We have seen that it is not true that only first-order quantifiers are recognized by finite state machines: If we accept *an even number of* as a quantifier, it is not first-order definable, but, of course, recognized by a finite state automaton.

$\mathbf{r}_{\text{AN EVEN NUMBER OF}} = \{<n,m>: n \text{ is even}\}$.



> A finite state automaton is **acyclic** if it contains no loops connecting two or more states.
>
> A finite state automaton M is **permutation invariant** iff for every two states S_i and S_j in M (and n the number of states), if $\alpha \in R^n_{i.,j}$ then $\pi(\alpha) \subseteq R^n_{i.,j}$.

Thus, permutation invariant means, that if α can be accepted between S_i and S_j than all permutations of α can. Permutation invariant automata recognize all and only permutation closed regular languages.

Van Benthem proves the following theorem:

Theorem: The first-order quantifiers are exactly the quantifiers that are recognized by permutation invariant acyclic finite state automata.

Proof: omitted.

As can be seen from the list, of r_{DET} above, the arithmetic relations that these quantifiers express are very simple: if $n = |X\cap Y|$ and $m = |X-Y|$ then we get:

r_{EVERY}	=	m=0
r_{SOME}	=	$n\neq 0$
r_{NO}	=	n=0
$r_{\text{AT LEAST 3}}$	=	$n\geq 3$
$r_{\text{AT MOST 3:}}$	=	$n\leq 3$
$r_{\text{EXACTLY k:}}$	=	n=3
$r_{\text{BETWEEN 2 AND 4}}$	=	$2 \leq n \leq 4$
r_{MOST} :	=	n>m

$r_{\text{AN EVEN NUMBER OF}} = \exists k[n = k + k]$

We can define *most* as: $n > m$ iff $\exists k[k \neq 0 \wedge n = m+k]$

The generalization is: all these expressions can be seen as first-order formulas in a language with constants n and m and addition +: **first-order additive arithmetics.**

$$(i + n \times a_1 + \ldots k \times a_m = i + \underbrace{a_1 + \ldots + a_1}_{\text{n times}} + \ldots + \underbrace{a_m + \ldots + am}_{\text{m times}}, \text{ which is additive})$$

Theorem: Every quantifier r_{DET} whose corresponding π-{0,1} language is context free is first-order additively definable.

Proof:
This follows from Parikh's theorem. Every context free language is semi-linear, hence a disjunction of linear languages. Each of the disjuncts is linear, which means that its length profile is of the form $i + n_1.a_1 + n_m.a_m$. Each such length-profile is an expression of first-order additive arithmetics. In other words: all semi-linear length profiles correspond to first-order additively definable k-place relations. Since all relevant languages are permutation closed, the behaviour of the quantifier corresponds directly to the behaviour of the length profiles. This means that the quantifier itself is definable in terms the union of the linear length profiles, and hence first-order additively definable.

All semi-linear sets correspond to first-order additively definable k-place relations. Ginsburg and Spanier proved in 1966 that for one place predicates and two place predicates the inverse also holds:

Theorem: The one place predicates and two place relations that are definable in first-order additive arithmetics are exactly the semilinear sets in a one-symbol ` – two-symbol alphabet.
Proof: Difficult.

van Benthem proves the following theorem:

Theorem: The first-order additively definable quantifiers are exactly the quantifiers whose corresponding π-{0,1} language is context free.

First-order additive arithmetics is a fragment of arithmetics which is complete and decidable. The incompleteness and undecidability come in with multiplication. Given this theorem, it seems that standard natural language quantifiers, including ones like *most* that are not first-order definable, are definable in first-order additive arithmetics, and hence stay below the Gödel-boundary.

The theorem follows as a corrollary from the following theorem that van Benthem proves:

Theorem: Every semi-linear π-{0,1} language is context free.

[Note that the restriction to a two-symbol alphabet is crucial. We saw before the language MIX = $\pi((abc)^*)$.]

Proof:
Since the union of context free languages is context free, you only need to prove that every **linear** π-{0,1} language is context free.
Let L be such a language. We need a pushdown storage automaton that will accept every string in L.
For each such string α, $\psi_\Sigma(\alpha) = <i_1,i_2> + n_1<a_{11},a_{12}> + \ldots + n_k<a_{m1},a_{m2}>$,
with $<i_1,i_2> \in I$ and $<a_{11},a_{12}> \ldots <a_{m1},a_{m2}> \in A$, where the first element of the pairs counts 0s while the second element of the pairs counts 1s.
If we can define a push-down storage automaton that computes these additions correctly, that automaton will recognize the strings in the language. This is, because the language is permutation closed, which means that recognizing the length-characteristics is sufficient to recognize the language.

Van Benthem defines such a pushdown automaton.

-The states are determined by the maximal number k among the i_1,i_2, $a_{11} \ldots a_{m2}$ in $I \cup A$: the states are all pairs $<i,j>$, $<i,j>^\#$ such that $i,j \leq k$.
You start in state $<0,0>$ and end in $<0,0>^\#$ with empty stack.

The trick about the automaton is that you can encode the number of 0s read as a progression of state transitions: $<0,j> \to <1,j> \to <2,j> \to \ldots$, and the same for the number of 1s: $<i,0> \to <i,1> \to <i,2> \to \ldots$,
and you can stack 0s and 1s on the stack without changing state. So, while you move for 0s, you can stack 1s.
And, you can at any point switch between these perspectives:
-loading 1s on the stack, by counting the relevant number of states back (on the second argument of the state-pairs),
-taking 1s off the stack, by counting the relevant number of states forward.

With only 2 symbols to keep track of, this creates enough computational power to prove the result.

Definition of the Pushdown Automaton:

Reading rules:

$i < k$	$(0,<i,j>,e) \to (<i+1,j>,e)$	$(0,<i,j>,e) \to (<i,j>,0)$
k		$(0,<k,j>,e) \to (<k,j>,0)$
$j < k$	$(1,<i,j>,e) \to (<i,j+1>,e)$	$(1,<i,j>,e) \to (<i,j>,1)$
k		$(1,<i,k>,e) \to (<i,k>,1)$

Empty rules:

	$(e,<i+1,j>,e) \to (<i,j>,0)$
$i<k$	$(e,<i,j>,0) \to (<i+1,j>,\sigma)$
	$(e,<i,j+1>,e) \to (<i,j>,1)$
$j<k$	$(e,<i,j>,1) \to (<i,j+1>,\sigma)$

Lowering rules:
Let $<m_1,m_2>$ be one of the pairs in $I \cup A$
and let $m_1 \leq i$ and $m_2 \leq j$:

$(e,<i,j>,e> \rightarrow (<i-m_1,j-m_2>,e)$

All these rules are the same for $^{\#}$-states:
Reading rules:

$i < k$	$(0,<i,j>^{\#},e) \rightarrow (<i+1,j>^{\#},e)$	$(0,<i,j>^{\#},e) \rightarrow (<i,j>^{\#},0)$
k		$(0,<k,j>^{\#},e) \rightarrow (<k,j>^{\#},0)$
$j < k$	$(1,<i,j>^{\#},e) \rightarrow (<i,j+1>^{\#},e)$	$(1,<i,j>^{\#},e) \rightarrow (<i,j>^{\#},1)$
k		$(1,<i,k>^{\#},e) \rightarrow (<i,k>^{\#},1)$

Empty rules:

	$(e,<i+1,j>^{\#},e) \rightarrow (<i,j>^{\#},0)$
$i<k$	$(e,<i,j>^{\#},0) \rightarrow (<i+1,j>^{\#},\sigma)$
	$(e,<i,j+1>^{\#},e) \rightarrow (<i,j>^{\#},1)$
$j<k$	$(e,<i,j>^{\#},1) \rightarrow (<i,j+1>^{\#},\sigma)$

Lowering rules:
Let $<m_1,m_2>$ be one of the pairs in $I \cup A$
and let $m_1 \leq i$ and $m_2 \leq j$:

$(e,<i,j>^{\#},e> \rightarrow (<i-m_1,j-m_2>^{\#},e)$

Crossing:
Let $<m_1,m_2>$ be one of the pairs in $I \cup A$
and let $m_1 \leq i$ and $m_2 \leq j$:

$(e,<i,j>,e) \rightarrow (<i-m_1,j-m_2>^{\#},e)$

van Benthem argues that you can prove, by inspecting the rules of the automaton, that the following holds:

Claim 1:
-at each stage in the computation in state $<i,j>$ there are numbers $x_1 \dots x_m$ such that:

i + the number of symbols 0 in the stack at $<i,j>$ =
the number of 0's read $- x_1.a_{11} - \dots - x_m.\ a_{m1}$

and

j + the number of symbols 1 in the stack at $<i,j>$ = =
the number of 1's read $- x_1.a_{12} - \dots - x_m.\ a_{m2}$

Similarly:
-at each stage in the computation in state $<i,j>^{\#}$ there are numbers $x_1 \dots x_m$ such that

i + the number of symbols 0 in the stack at $<i,j>$ =
the number of 0's read $- i_1 - x_1.a_{11} - \dots - x_m.\ a_{m1}$

and

j + the number of symbols 1 in the stack at $<i,j>$ =
the number of 1's read $- i_2 - x_1.a_{12} - \dots - x_m.\ a_{m2}$

(Here $<i_1,i_2> \in I$, so these numbers have nothing to do with i in $<i,j>$.)

This means that in <0,0> with empty stack:

the number of 0's read $- x_1.a_{11} - \ldots - x_m.\ a_{m1} = 0$

and the number of 1's read $- x_1.a_{12} - \ldots - x_m.\ a_{m2} = 0$

hence:

in <0,0> with empty stack:

the number of 0's read = $x_1.a_{11} + \ldots + x_m.\ a_{m1}$

the number of 1's read = $x_1.a_{12} + \ldots + x_m.\ a_{m2} = 0$

and in $<0,0>^{\#}$ with empty stack:

the number of 0's read = $i_1 + x_1.a_{11} + \ldots + x_m.\ a_{m1}$

the number of 1's read = $i_2 + x_1.a_{12} + \ldots + x_m.\ a_{m2} = 0$

This means that indeed only for α such that for some $i, a_{1\ldots,}a_m, n_1\ldots n_m$:
$\psi_\Sigma(\alpha) = i + x_1.a_1 + \ldots + x_m.\ a_m$ does the computation get to $<0,0>^{\#}$.

This means that the language recognized is a subset of L.

Conversely, we need to prove that **every** string in L is recognized.

Claim 2:

-If the automaton is in state <0,0> and the stack consists of 0s only or of 1s only
and b_1 = the number of 0s in the stack + the number of 0s still to read
and b_2 = the number of 1s in the stack + the number of 1s still to read
and $<b_1,b_2> = <i_1,i_2> + n_1<a_{11},a_{12}> + \ldots + n_k<a_{m1},a_{m2}>$,
then the automaton will continue to recognize the string.

-If the automaton is in state $<0,0>_{\#}$ and the stack consists of 0s only or of 1s only
and b_1 = the number of 0s in the stack + the number of 0s still to read
and b_2 = the number of 1s in the stack + the number of 1s still to read
and $<b_1,b_2> = n_1<a_{11},a_{12}> + \ldots + n_k<a_{m1},a_{m2}>$,
then the automaton will continue to recognize the string.

From the fact that L is semi-linear and the stack is empty in <0,0> at the beginning of the computation for $\alpha \in L$, if the claim is true, it follows that the automaton will recognize α, and with that, the theorem.

Proof of the claim: Even with the appendix, this proof is hard to follow.

BIBLIOGRAPHY

Two classical textbooks on Formal Language Theory:

Hopcroft, John and Jeffrey Ullmann, 1979, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, Massachusets.

Lewis, Harry and Christos Papadimitriou, 1981, *Elements of the Theory of Computation*, Prentic Hall, New Jersey.

A critical discussion of all arguments for the non-contextfreeness of natural languages in the literature up to 1982:

Pullum, Geoffrey and Gerald Gazdar, 1982, 'Natural languages and contextfree languages,' in: *Linguistics and Philosphy* 4.

The arguments from Bambara, Mandarin Chinese, and Swiss German:

Culy, Chris, 1985, 'The complexity of the vocabulary of Bambara,' in: *Linguistics and Philosophy* 8.

Radzinsky, Daniel, 1990, 'Unbounded syntactic copying in Mandarine Chinese,' in: *Linguistics and Philosophy* 13.

Shieber, Stuart, 1985, 'Evidence against the contextfreeness of natural languages,' in: *Linguistics and Philosophy* 8.

The parsing of the verb cluster in Dutch and German:

Bach, Emmon, Colin Brown and William Marslen-Wilson, 1986, 'Crossed and nested dependencies in Dutch and German,' in: *Language and Cognitive Processes* 1.

Joshi, Aravind, 1989, 'Processing crossed and nested dependencies,' in: *Language and Cognitive Processes* 3.

Tree adjoining grammars:

Joshi, Aravind, 1983, 'How much context sensitivity is necessary for characterizing structural descriptions,' in: Dowty et. al. (eds.) *Natural Language Processing. Theoretical, Computational and Psychological Perspectives*, Cambridge University Press, Cambridge.

Joshi, Aravind, 1987, 'An introduction to tree adjoining grammars,' in: Manaster-Ramer (ed.), *Mathematics of Language*, Benjamins, Amsterdam.

Kroch, Antony, 1987, 'Unbounded dependencies and Subjacency in a tree adjoining grammar,' in: Manaster-Ramer (ed.), *Mathematics of Language*, Benjamins, Amsterdam.

VijayShanker, K., 1988, *A Study of Tree Adjoining Grammars*, Report MS-CIS-88-03 LINC LAB 95, Department of Computer Science, University of Pennsylvania.

MCTags and Linear Context-free Rewriting Systems:

VijayShanker,K., Weir, D., and A. Joshi, 1987, 'On the progression from context free to tree adjoining grammars,' in: Manaster-Ramer (ed.), *Mathematics of Language*, Benjamins, Amsterdam.

Kallmeyer, 2013, 'Linear Context-Free Rewriting Systems,' in: Language and Linguiscs Compass 7,1 pp. 22-38

Weir, David, 1987, *From Contextfree Grammars to Tree Adjoining Grammars and Beyond*, Report MS-CIS-87-42, LIC LAB65, Department of Computer Science, University of Pennsylvania.

Beyond mildly contextsensitive

Manaster-Ramer, Alexis, 1987, 'Dutch as a Formal Language,' in: *Linguistics and Philosophy*, 10, 2, pp. 221-246.

Radzinski, Daniel, 1991, 'Chinese number-names, tree adjoining languages, and mild context-sensitivity,' in: *Computational Linguistics* 17, pp. 277–299.

Salvati, Silvain, 2015, MIX is a 2-MCFL and the word problem in Z^2 is solved by a third-order collapsible pushdown atomaton, Journal of Computer and System Sciences 81, pp. 1252-1277

Stabler, Edward, P., 2010, 'Computational perspectives on minimalism, in: C. Boeckx (ed), *Oxford Handbook* of Linguistic Minimalism, pp. 616=641, Oxford University Press, Oxford.

Categorial grammar.

On type shifting:

Partee, Barbara, 1987, 'Noun phrase interpretation and type-shifting principles,' in: Groenendijk, J, D. de Jongh, and M. Stokhof (eds.), *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers,* Foris, Dordrecht.

Landman, Fred, classnotes Advanced Semantics.

On combinatory categorial grammar:

Jacobson, Pauline, 1999, 'Towards variable-free semantics,' in: *Linguistics and Philosophy*, 22, pp. 117=184

Steedman, Mark, 1996, *Surface Structure and Interpretation,* The MIT Press.

Steedman, Mark, 2000, *The Syntactic Process,* The MIT Press.

Szabolcsi, Anna, 1989, 'Bound variables in syntax (are there any?),' in: Bartsch, R, J. van Benthem and P. van Emde-Boas, *Semantics and Contextual Expression*, pp. 294-318, Foris, Dordrecht.

Proof-theoretic grammar

Lambek, Joachim, 1958, 'The mathematics of sentence structure,' in: American Mathematical Monthly 65,pp. 154 - 170

Moortgat, Michael, 1997, 'Categorial type logics,' in: van Benthem, J. and A, ter Meulen (eds.) *Handbook of Logic and Languages,* ch 2.

Semantic Automata:

van Benthem, Johan, 1983, ‘Determiners and Logic,’ in: *Linguistics and Philosophy* 6.

van Benthem, Johan, 1986, *Essays in Logical Semantics*, Kluwer, Dordrecht.

van Benthem, Johan, 1987, ‘Semantic automata,’ in: Groenendijk, de Jongh and Stokhof (eds.) *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers*, Foris, Dordrecht.