How can we reason/compute with natural language?

There’s an awful lot of it around (e.g. in the web)

▶ Semantic Web: AI and the future of the WWW
▶ Ontologies: organizing knowledge via categories and relations
▶ Description Logics
▶ Finite State Methods for Reasoning about Change/Time
How can we reason/compute with natural language?

There's an awful lot of it around (e.g. in the web)

- Semantic Web: AI and the future of the WWW
- Ontologies: organizing knowledge via categories and relations
- Description Logics
- Finite State Methods for Reasoning about Change/Time
How can we reason/compute with natural language?
There's an awful lot of it around (e.g. in the web)
Q  *How can we reason/compute with natural language?*
There’s an awful lot of it around (e.g. in the web)

- Semantic Web: AI and the future of the WWW
Q  *How can we reason/compute with natural language?*
There’s an awful lot of it around (e.g. in the web)

- Semantic Web: AI and the future of the WWW
- Ontologies: organizing knowledge via categories and relations
How can we reason/compute with natural language?
There’s an awful lot of it around (e.g. in the web)

- Semantic Web: AI and the future of the WWW
- Ontologies: organizing knowledge via categories and relations
- Description Logics
How can we reason/compute with natural language?
There’s an awful lot of it around (e.g. in the web)

- Semantic Web: AI and the future of the WWW
- Ontologies: organizing knowledge via categories and relations
- Description Logics
- Finite State Methods for Reasoning about Change/Time
Tolerance and Sorites chains

A unary relation $P$ is *tolerant up to near*$_P$ if

$$\text{near}_P(x, y) \implies (P(x) \implies P(y)).$$

**Example 1.** $P(x)$ is $\text{heap}(x)$,

near$_P(x, y)$ is $|x - y| \leq 1$ grain
A unary relation $P$ is tolerant up to $\text{near}_P$ if

$$\text{near}_P(x, y) \rightarrow (P(x) \rightarrow P(y)).$$

**Example 1.** $P(x)$ is $\text{heap}(x)$,  
$\text{near}_P(x, y)$ is $|x - y| \leq 1$ grain

**Example 2.** $P(x)$ is $\text{walking-distance}(x)$,  
$\text{near}_P(x, y)$ is $|x - y| \leq 1$ foot
Tolerance and Sorites chains

A unary relation $P$ is tolerant up to $\text{near}_P$ if

$$\text{near}_P(x, y) \rightarrow (P(x) \rightarrow P(y)).$$

**Example 1.** $P(x)$ is $\text{heap}(x)$,

$\text{near}_P(x, y)$ is $|x - y| \leq 1$ grain

**Example 2.** $P(x)$ is $\text{walking-distance}(x)$,

$\text{near}_P(x, y)$ is $|x - y| \leq 1$ foot

**Example 3.** $P(x)$ is $\text{young}(x)$, $\text{sunny}(x)$,

$\text{near}_P(x, y)$ is $|x - y| \leq 1$ picosec
A unary relation $P$ is *tolerant up to near* $P$ if
\[ \text{near}_P(x, y) \rightarrow (P(x) \rightarrow P(y)). \]

**Example 1.** $P(x)$ is *heap*(x),
\[ \text{near}_P(x, y) \text{ is } |x - y| \leq 1 \text{ grain} \]

**Example 2.** $P(x)$ is *walking-distance*(x),
\[ \text{near}_P(x, y) \text{ is } |x - y| \leq 1 \text{ foot} \]

**Example 3.** $P(x)$ is *young*(x), *sunny*(x),
\[ \text{near}_P(x, y) \text{ is } |x - y| \leq 1 \text{ picosec} \]

A *Sorites chain* is a sequence $y_1, \ldots, y_n$ such that $P$ holds of $y_1$ but not $y_n$, even though $\text{near}_P(y_i, y_{i+1})$ for $1 \leq i < n$. 
Regular expressions $\equiv$ MSO (Büchi-Elgot-Trakhtenbrot)

Regular languages $\subseteq \Sigma^+ = \text{sets of strings definable in } \text{MSO}_\Sigma$
Regular expressions $= \text{MSO (Büchi-Elgot-Trakhtenbrot)}$

Regular languages $\subseteq \Sigma^+ = \text{sets of strings definable in MSO}_\Sigma$

$\text{MSO}_\Sigma = \text{monadic second-order logic with}$

a binary relation symbol (successor) and

a unary relation symbol for each symbol in $\Sigma$

$abbc \leadsto \text{MSO}_\Sigma$-model $\langle \{1, 2, 3, 4\}, S_4, U_a, U_b, U_c \rangle$
Regular expressions $= \text{MSO (Büchi-Elgot-Trakhtenbrot)}$

Regular languages $\subseteq \Sigma^+ = \text{sets of strings definable in MSO}_\Sigma$

$\text{MSO}_\Sigma = \text{monadic second-order logic with}$
$\text{a binary relation symbol (successor) and}$
$\text{a unary relation symbol for each symbol in } \Sigma$

$\text{abbc } \rightsquigarrow \text{MSO}_\Sigma\text{-model } \langle \{1,2,3,4\}, S_4, U_a, U_b, U_c \rangle \text{ where}$
$S_4 := \{ \langle 1,2 \rangle, \langle 2,3 \rangle, \langle 3,4 \rangle \}$
Regular expressions = MSO (Büchi-Elgot-Trakhtenbrot)

Regular languages $\subseteq \Sigma^+ = \text{sets of strings definable in } \text{MSO}_\Sigma$

$\text{MSO}_\Sigma = \text{monadic second-order logic with}$

a binary relation symbol (successor) and

a unary relation symbol for each symbol in $\Sigma$

$abbc \leadsto \text{MSO}_\Sigma$-model $\langle \{1, 2, 3, 4\}, S_4, U_a, U_b, U_c \rangle$ where

$S_4 := \{ \langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle \}$

$U_a := \{ 1 \}$
Regular expressions = MSO (Büchi-Elgot-Trakhtenbrot)

Regular languages \( \subseteq \Sigma^+ \) = sets of strings definable in MSO\(_\Sigma\)

MSO\(_\Sigma\) = monadic second-order logic with
a binary relation symbol (successor) and
a unary relation symbol for each symbol in \( \Sigma \)

\( a bbcc \sim \) MSO\(_\Sigma\)-model \( \langle \{1, 2, 3, 4\}, S_4, U_a, U_b, U_c \rangle \) where
\( S_4 := \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 4\rangle\} \)
\( U_a := \{1\} \)
\( U_b := \{2, 3\} \)
Regular expressions $= \text{MSO (Büchi-Elgot-Trakhtenbrot)}$

Regular languages $\subseteq \Sigma^+ =$ sets of strings definable in $\text{MSO}_\Sigma$

$\text{MSO}_\Sigma =$ monadic second-order logic with

- a binary relation symbol (successor) and
- a unary relation symbol for each symbol in $\Sigma$

$\text{abbc} \leadsto \text{MSO}_\Sigma$-model $\langle \{1, 2, 3, 4\}, S_4, U_a, U_b, U_c \rangle$ where

- $S_4 := \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle\}$
- $U_a := \{1\}$
- $U_b := \{2, 3\}$
- $U_c := \{4\}$
Regular expressions $= \text{MSO (B"{u}chi-Elgot-Trakhtenbrot)}$

Regular languages $\subseteq \Sigma^+ = \text{sets of strings definable in MSO}_\Sigma$

$\text{MSO}_\Sigma = \text{monadic second-order logic with}$

a binary relation symbol (successor) and

a unary relation symbol for each symbol in $\Sigma$

$abbc \mapsto \text{MSO}_\Sigma$-model $\langle \{1, 2, 3, 4\}, S_4, U_a, U_b, U_c \rangle$ where

$S_4 := \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle\}$

$U_a := \{1\}$

$U_b := \{2, 3\}$

$U_c := \{4\}$

$(\exists x_1)(\exists x_2)(\exists x_3)(\exists x_4)$ $U_a(x_1) \land U_b(x_2) \land U_b(x_3) \land U_c(x_4) \land$

$S(x_1, x_2) \land S(x_2, x_3) \land S(x_3, x_4)$
Turing Awards

1996  Amir Pnueli

For seminal work introducing temporal logic into computing science and for outstanding contributions to program and systems verification.
1996  Amir Pnueli

For seminal work introducing temporal logic into computing science and for outstanding contributions to program and systems verification.

2007  Edmund M. Clarke, E. Allen Emerson and Joseph Sifakis

For their roles in developing Model-Checking into a highly effective verification technology, widely adopted in the hardware and software industries.
1996  Amir Pnueli

For seminal work introducing temporal logic into computing science and for outstanding contributions to program and systems verification.

2007  Edmund M. Clarke, E. Allen Emerson and Joseph Sifakis

For their roles in developing Model-Checking into a highly effective verification technology, widely adopted in the hardware and software industries.

2011  Judea Pearl

For fundamental contributions to artificial intelligence through the development of a calculus for probabilistic and causal reasoning.