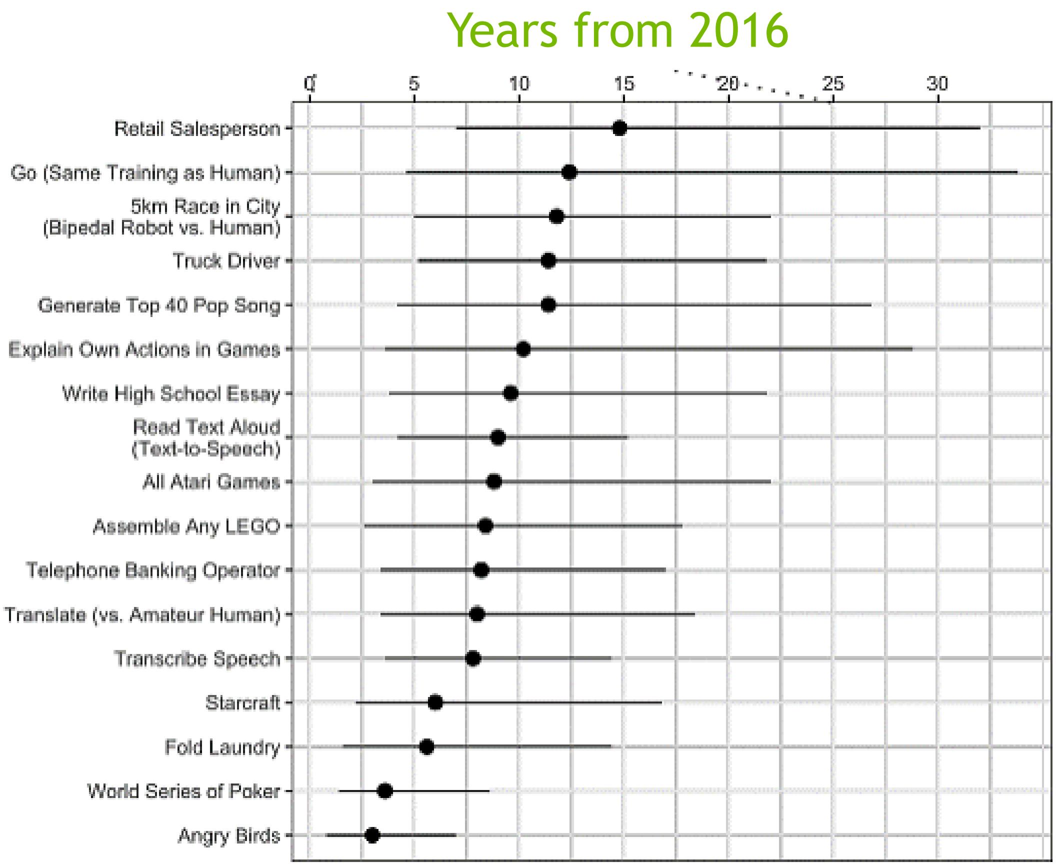
MACHINE REASONING: A PERSPECTIVE AND POSSIBILITY

Aik Beng NG, Zhangsheng LAI, Mar 18

AI EXCEEDING HUMAN PERFORMANCE

Timeline Estimates for AI Achieving Human Performance[1]

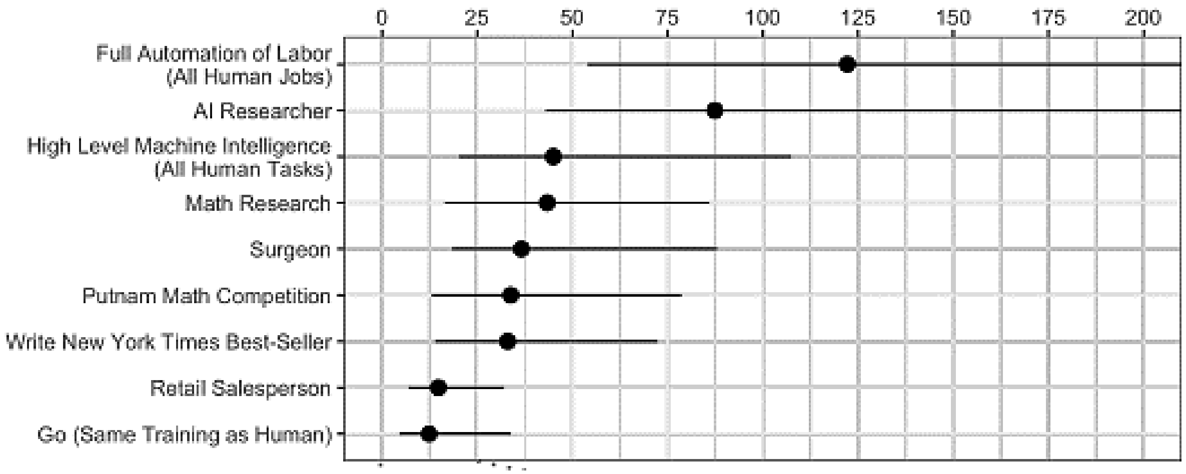


[1] Grace et al. "When Will AI Exceed Human Performance? Evidence from AI Experts" Journal of Artificial Intelligence Research 62, 2018, 729-754

AI EXCEEDING HUMAN PERFORMANCE

Timeline Estimates for AI Achieving Human Performance[1]

Years from 2016

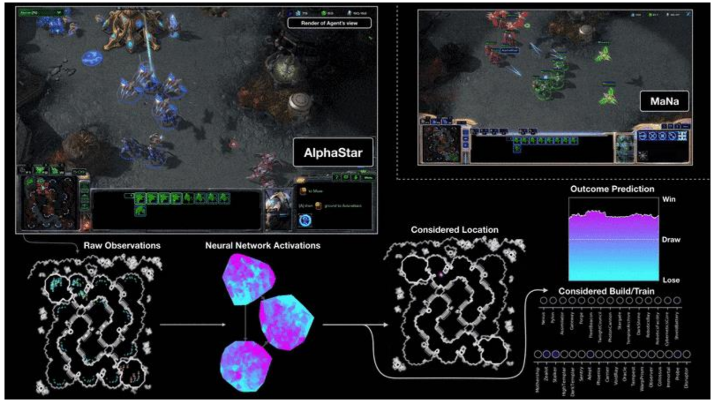


[1] Grace et al. "When Will AI Exceed Human Performance? Evidence from AI Experts" Journal of Artificial Intelligence Research 62, 2018, 729-754

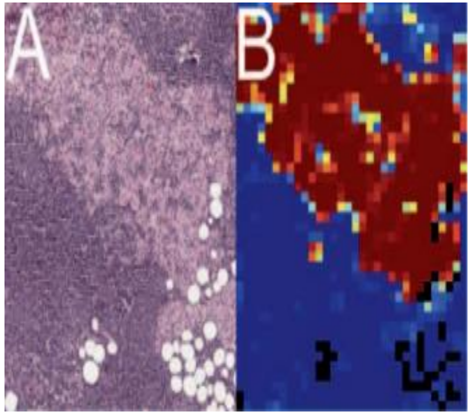
WHERE ARE WE TODAY?

Image Recognition[2]

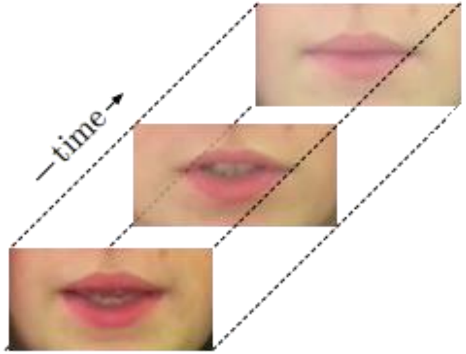
Face Recognition[3]



Starcraft II[4]



Cancer Detection[5]



Lip Reading[6]

[2] He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". ICCV '15 Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), 2015, Pages 1026-1034
[3] Chaochao Lu and Xiaoou Tang. "Surpassing Human-Level Face Verification Performance on LFW with GaussianFace". AAAI'15 Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, Pages 3811-3819
[4] The Challenge of StarCraft, DeepMind
[5] Liu et al. "Artificial Intelligence-Based Breast Cancer Nodal Metastasis Detection". Archives of Pathology & Laboratory Medicine In-Press., 2018
[6] Assael et al. "LipNet: End-to-End Sentence-level Lipreading". arXiv:1611.01599v2 [cs.LG], 2016

AI-ENABLED RECOGNITION

Significant AI ability underlying many AI successes

"to identify something from prior knowledge"

Re cognition

AI-ENABLED RECOGNITION

Significant AI ability underlying many AI successes

“to identify something from prior knowledge”

Re cognition

“again”, “once more”

“understand through thought, experience, senses”,
“to acquire knowledge”

AI-ENABLED RECOGNITION

Significant AI ability underlying many AI successes

"to identify something from prior knowledge"

Re cognition

AI-ENABLED RECOGNITION

Significant AI ability underlying many AI successes

Re cognition

AI-ENABLED RECOGNITION

Significant AI ability underlying many AI successes

Re cognition

"again", "once more"

"understand through thought, experience, senses",
"to acquire knowledge"

AI-ENABLED RECOGNITION

Significant AI ability underlying many AI successes

“to identify something from prior knowledge”

Re cognition

“again”, “once more”

“understand through thought, experience, senses”, “to acquire knowledge”

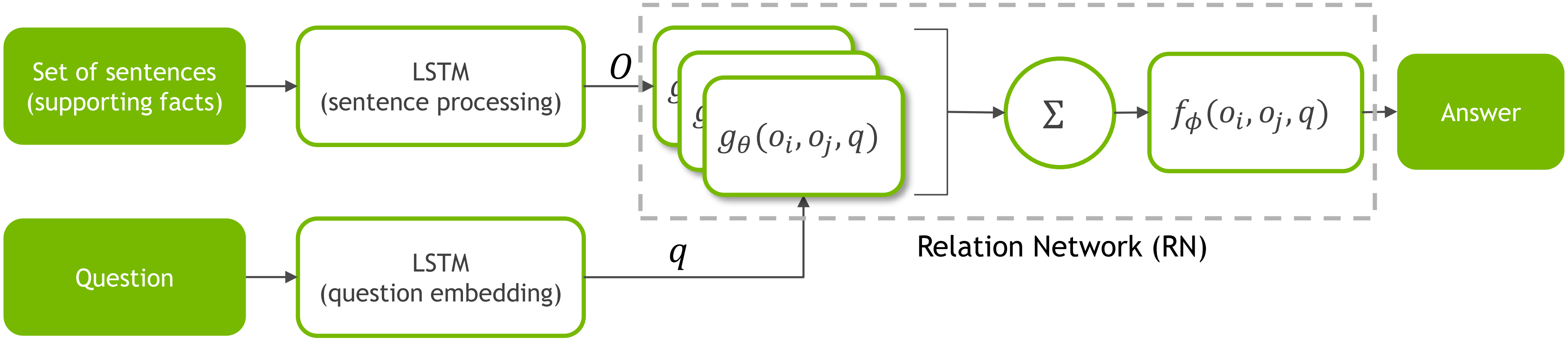
REASONING. A KEY ASPECT OF COGNITION

"A plausible definition of 'reasoning' could be 'algebraically manipulating previously acquired knowledge in order to answer a new question'." [7]

[7] Leon Bottou. "From machine learning to machine reasoning". Machine Learning Volume 94 Issue 2, 2004, Pages 133-149

SIMPLE NEURAL NETWORK MODULE FOR RELATIONAL REASONING[8]

Reasoning about relations between "objects"



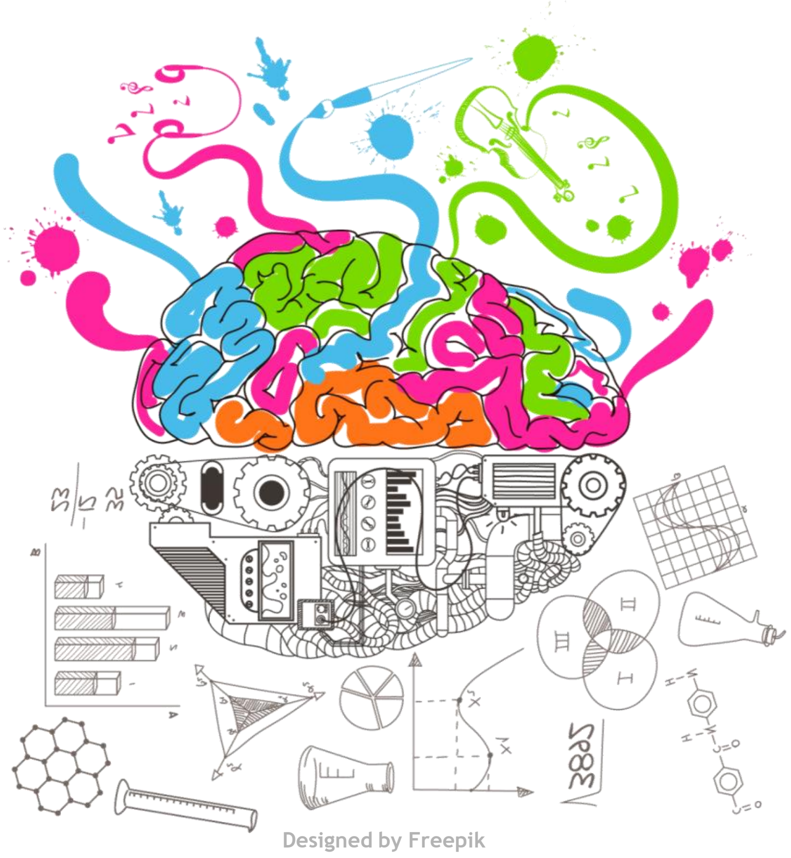
$$RN(O) = f_\phi\left(\sum_{i,j} g_\theta(o_i, o_j, q)\right)$$

where input is set of objects $O = \{o_1, o_2, \dots, o_n\}$ *and question embedding* q
and f_ϕ *and* g_θ *are MLPs*

[8] Adam Santoro, David Raposo, David G. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Tim Lillicrap. "A simple neural network module for relational reasoning." In Advances in neural information processing systems, pp. 4974-4983, 2017.

ADVANCING REASONING

Theory of 2 distinct types of reasoning[9] has long existed



System 1
(also known as Type 1)

"Fast and intuitionistic thinking"

- Rapid, automatic, unconscious.
- Involves prior knowledge, beliefs, heuristics.
- Instinctive behaviours innately programmed.

System 2
(also known as Type 2)

"Slow and deliberate thinking"

- Slow, sequential, conscious.
- Capable of abstract and hypothetical thinking.
- Support decisions by constructing mental models or simulations of future possibilities.

[9] Jonathan St. B.T. Evans. "In two minds: dual-process accounts of reasoning". Trends in cognitive sciences 7, no. 10, 2003, 454-459

CONSIDER THIS:

$$\begin{bmatrix} \mathbf{1} & \mathbf{2} \\ \mathbf{3} & \mathbf{4} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{4} & \mathbf{3} \\ \mathbf{2} & \mathbf{1} \end{bmatrix} = ?$$

CONSIDER THIS:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$$

"Fast and intuitionistic thinking"

- Math problem, specifically matrix operations.
- Multiplication and addition.
- Approximate sense of values within the resulting matrix.

"Slow and deliberate thinking"

- Enters into analytical thinking.
- Performs precise steps to derive answer.

CONSIDER THIS:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} = \cancel{\begin{bmatrix} 4 \\ 4 \end{bmatrix}} \begin{bmatrix} 8 & 5 \\ 20 & 13 \end{bmatrix}$$

“Fast and intuitionistic thinking”

- Math problem, specifically matrix operations.
- Multiplication and addition.
- Approximate sense of values within the resulting matrix.
- **2 x 2 resulting matrix!**

“Slow and deliberate thinking”

- Enters into analytical thinking.
- Performs precise steps to derive answer.

“A plausible definition of ‘reasoning’ could be ‘algebraically manipulating previously acquired knowledge in order to answer a new question’.” [7]

Natural Representations

Modular and Composable

Constructive

TYPE THEORY

It all begin from Russell's Paradox

Type theory is a branch of mathematical symbolic logic that formalizes the idea that each term if of some definitive *type*.

We write $a : A$ which can be interpreted in two ways:

- The **term** a is of type A
- a is a **proof** of proposition A

$$2019 : \mathbb{N}$$

$$[1; 0.75; 2.3; 18.3] : \mathrm{Vec}(\mathbb{R}, 4)$$

```
Lemma simple : forall (n : nat), n = n.
Proof. intros. reflexivity. Qed.
simple : forall (n : nat), n = n.

Lemma impossible : forall (n : nat), n = n+1.
?? : forall (n : nat), n = n+1.
```

DEPENDENT TYPES

Types that depend on a term or another type

Dependent pair types ($\sum$-types) are types with two components where the type of the second component is allowed to vary depending on the choice of the first component.

The dependent pair type is written as $\sum_{(x:A)} B(x)$ with term $(a, b) : \sum_{(x:A)} B(x)$, given $a : A$ and $b : B(a)$.

$$\sum_{(c:Color)} Fruits(c)$$

(red, apple) : $\sum_{(c:Color)} Fruits(c)$
(silver, ??) : $\sum_{(c:Color)} Fruits(c)$

projT1 (red, apple) = red
projT2 (red, apple) = apple

FAMILY RELATIONS

Who is the father?

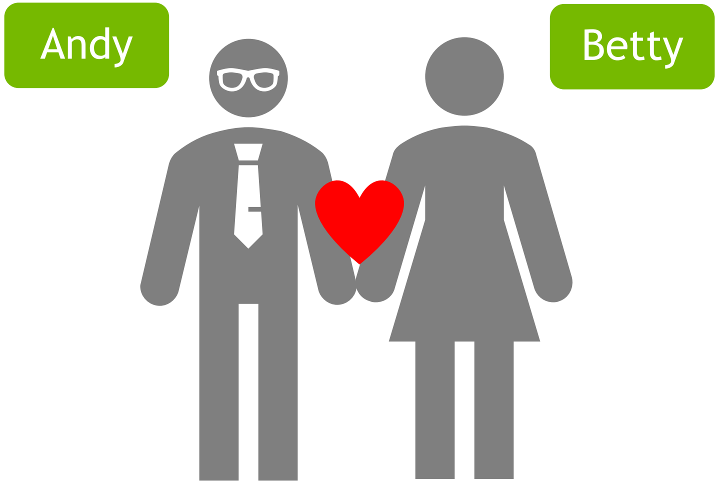
FAMILY RELATIONS

Who is the father?

(Betty, birthcert) : $\sum_{p:Person} Mother_{Tom}(p)$

FAMILY RELATIONS

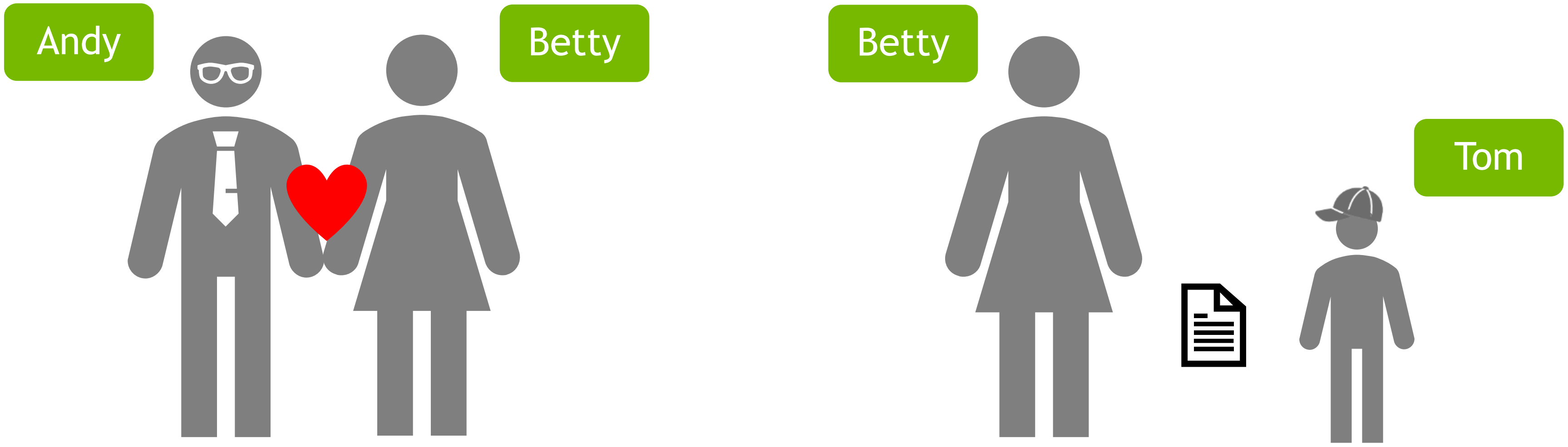
Who is the father?



(Andy, marriagecert) : $\sum_{p:People} Husband_{Betty}(p)$

FAMILY RELATIONS

Who is the father?



FAMILY RELATIONS

Who is the father?

$$(??,\ ??) : \sum_{p:People} Father_{Tom}(p)$$

FAMILY RELATIONS

Who is the father?

Mother's Husband is Father

FAMILY RELATIONS

Who is the father?

Mother’s Husband is Father

$$(??, ??) : \sum_{p:People} Father_{Tom}(p)$$

findFather : forall (x : Person) (y : $\mathbb{M}$), $\mathbb{H}$ -> Person
prfFather : forall (x : Person) (y : $\mathbb{M}$) (z : $\mathbb{H}$), $Father_x(projT1\ z)$

where

$$\mathbb{M} = \sum_{p:Person} Mother_x(p)$$

$$\mathbb{H} = \sum_{p:Person} Husband_{projT1\ (y)}(p)$$

FAMILY RELATIONS

Who is the father?

Mother's Husband is Father

(??, ??) : $\sum_{p:People} Father_{Tom}(p)$

Merge findFather and prfFather

infFather : forall (x : Person) (y : $\mathbb{M}$), $\mathbb{H}$ -> $\sum_{p:People} Father_{x}(p)$

where

$$\mathbb{M} = \sum_{p:Person} Mother_{x}(p)$$

$$\mathbb{H} = \sum_{p:Person} Husband_{projT1\,(y)}(p)$$

FAMILY RELATIONS

Who is the father?

Mother's Husband is Father

$$(??,\ ??) : \sum_{p:People} Father_{Tom}(p)$$

infFather : forall (x : Person) (y : $\mathbb{M}$), $\mathbb{H}$ -> $\sum_{p:People} Father_{x}(p)$

```
Theorem father_of_Tom : sigT (Father Tom).
Proof. simple refine (infFather _ _ _).
  exact (Betty, birthcert).
  exact (Andy, marriagecert). Defined.
```

FAMILY RELATIONS

Finding the father

Goal Window

1 subgoal
(Betty, birthcert) : $\sum_{p:Person} Mother_{Tom}(p)$
(Andy, marriagecert) : $\sum_{p:People} Husband_{Betty}(p)$
____(1/1)
$\sum_{p:People} Father_{Tom}(p)$

Proof Window

```
Theorem father_of_Tom : sigT (Father Tom).
Proof. simple refine (infFather _ _ _).
  exact (Betty, birthcert).
  exact (Andy, marriagecert). Defined.
```

FAMILY RELATIONS

Finding the father

Goal Window

2 subgoals

(Betty, birthcert) : $\sum_{p:Person} Mother_{Tom}(p)$

(Andy, marriagecert) : $\sum_{p:People} Husband_{Betty}(p)$

____(1/2)

$\sum_{p:Person} Mother_{Tom}(p)$

____(2/2)

$\sum_{p:People} Husband_{??}(p)$

Proof Window

```
Theorem father_of_Tom : sigT (Father Tom).
Proof. simple refine (infFather _ _ _).
  exact (Betty, birthcert).
  exact (Andy, marriagecert). Defined.
```

FAMILY RELATIONS

Finding the father

Goal Window

1 subgoal

(Betty, birthcert) : $\sum_{p:Person} Mother_{Tom}(p)$

(Andy, marriagecert) : $\sum_{p:People} Husband_{Betty}(p)$

____(1/1)

$\sum_{p:People} Husband_{Betty}(p)$

Proof Window

```
Theorem father_of_Tom : sigT (Father Tom).
Proof. simple refine (infFather _ _ _).
  exact (Betty, birthcert).
  exact (Andy, marriagecert). Defined.
```

FAMILY RELATIONS

Finding the father

Goal Window

No more subgoals.

Proof Window

```
Theorem father_of_Tom : sigT (Father Tom).
Proof. simple refine (infFather _ _ _).
  exact (Betty, birthcert).
  exact (Andy, marriagecert). Defined.
```

FAMILY RELATIONS

Finding the father

Goal Window

father_of_Tom is defined

Proof Window

```
Theorem father_of_Tom : sigT (Father Tom).
Proof. simple refine (infFather _ _ _).
  exact (Betty, birthcert).
  exact (Andy, marriagecert). Defined.
```

FAMILY RELATIONS

Who is the father?

We have constructed the term,

father_of_Tom = (Andy, prfFather Tom (Betty, birthcert) (Andy, marriagecert)) : $\sum_{p:People} Father_{Tom}(p)$

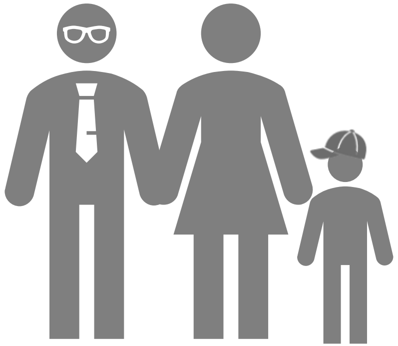
Using represented information

(Betty, birthcert) : $\sum_{p:Person} Mother_{Tom}(p)$

(Andy, marriagecert) : $\sum_{p:People} Husband_{Betty}(p)$

And the encoded knowledge

infFather : forall (x : Person) (y : $\mathbb{M}$), $\mathbb{H}$ -> $\sum_{p:People} Father_{x}(p)$



“A plausible definition of ‘reasoning’ could be ‘algebraically manipulating previously acquired knowledge in order to answer a new question’.” [7]

Natural Representations

Modular and Composable

Constructive

"A plausible definition of 'reasoning' could be 'algebraically manipulating previously acquired knowledge in order to answer a new question'." [7]

Natural Representations

Modular and Composable

Constructive

(Betty, birthcert) : $\sum_{p:Person} Mother_{Tom}(p)$

(Andy, marriagecert) : $\sum_{p:People} Husband_{Betty}(p)$

"A plausible definition of 'reasoning' could be 'algebraically manipulating previously acquired knowledge in order to answer a new question'." [7]

Natural Representations

Modular and Composable

Constructive

(Betty, birthcert) : $\sum_{p:Person} Mother_{Tom}(p)$

infFather : forall (x : Person) (y : $\mathbb{M}$),
$\mathbb{H}$ -> $\sum_{p:People} Father_{x}(p)$

“A plausible definition of ‘reasoning’ could be ‘algebraically manipulating previously acquired knowledge in order to answer a new question’.” [7]

Natural Representations

Modular and Composable

Constructive

father_of_Tom =
(Andy, prfFather Tom (Betty, birthcert) (Andy, marriagecert))
$: \sum_{p:People} Father_{Tom}(p)$

“A plausible definition of ‘reasoning’ could be ‘algebraically manipulating previously acquired knowledge in order to answer a new question’.” [7]

Natural Representations

Modular and Composable

Constructive

NVIDIA®