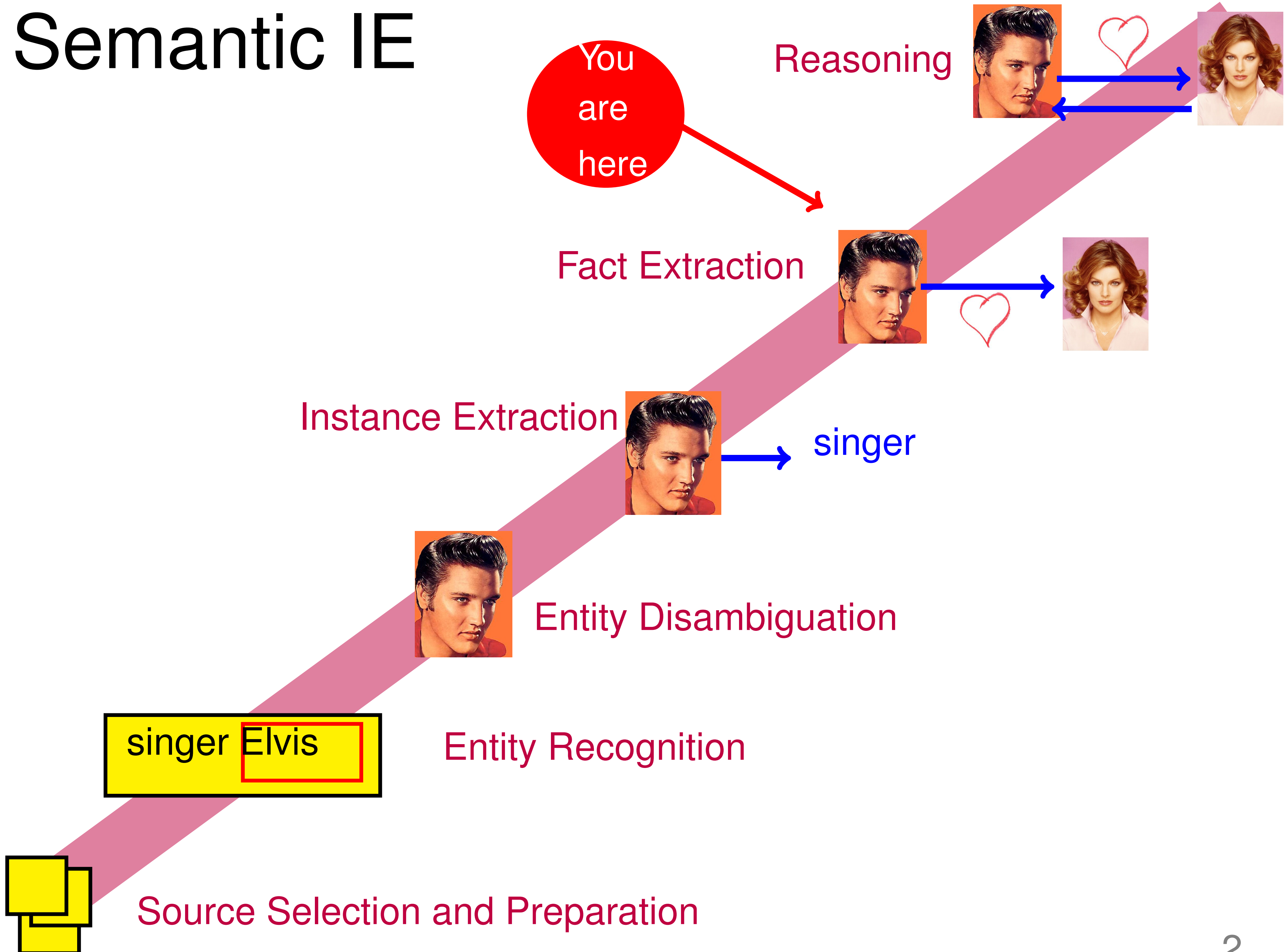
Information Extraction by Reasoning

Fabian M. Suchanek

Semantic IE



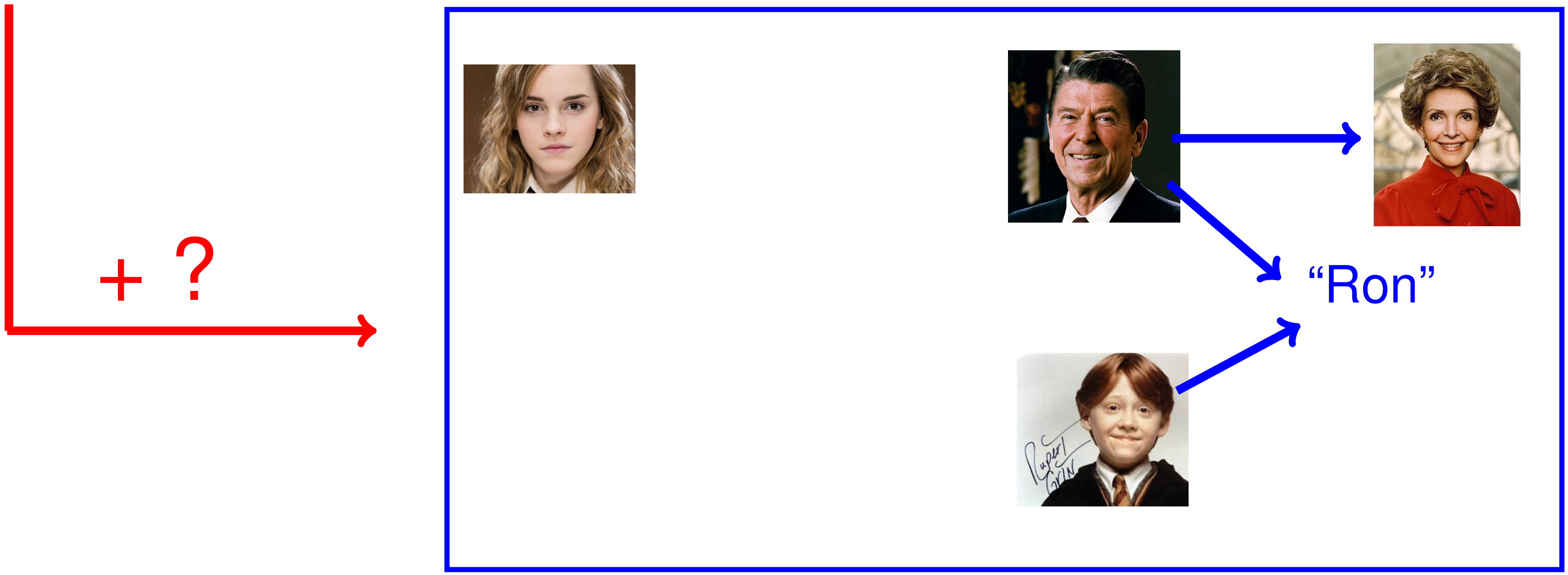
You are here
Reasoning
Fact Extraction
Instance Extraction
singer
Entity Disambiguation
singer Elvis
Entity Recognition
Source Selection and Preparation

Problems in extending a KB

“Hermione is married to Ron”

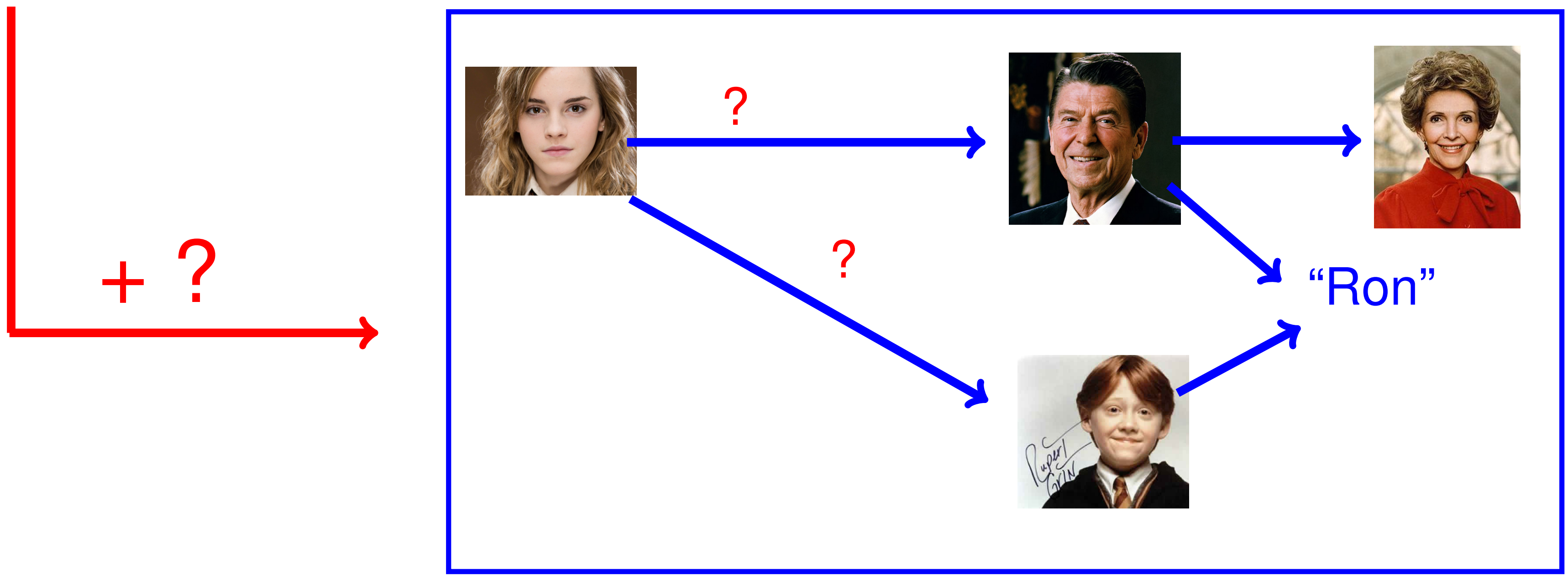
Problems in extending a KB

"Hermione is married to Ron"



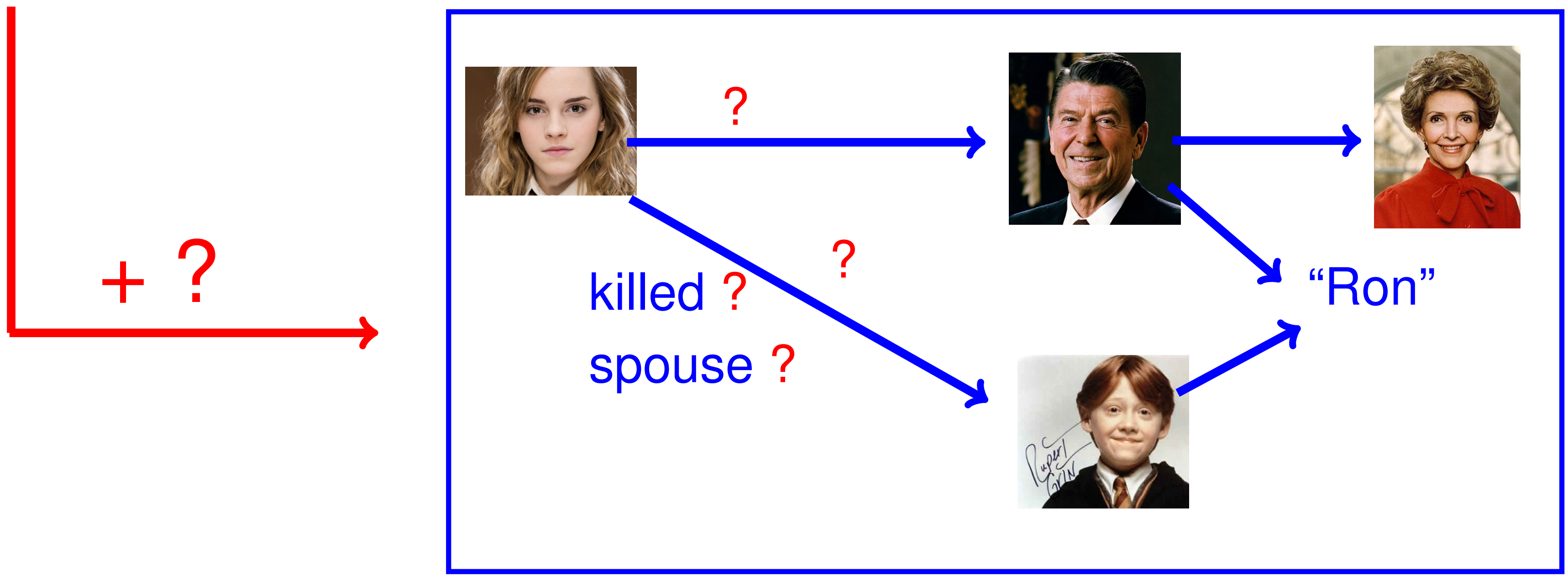
Problems in extending a KB

“Hermione is married to Ron”



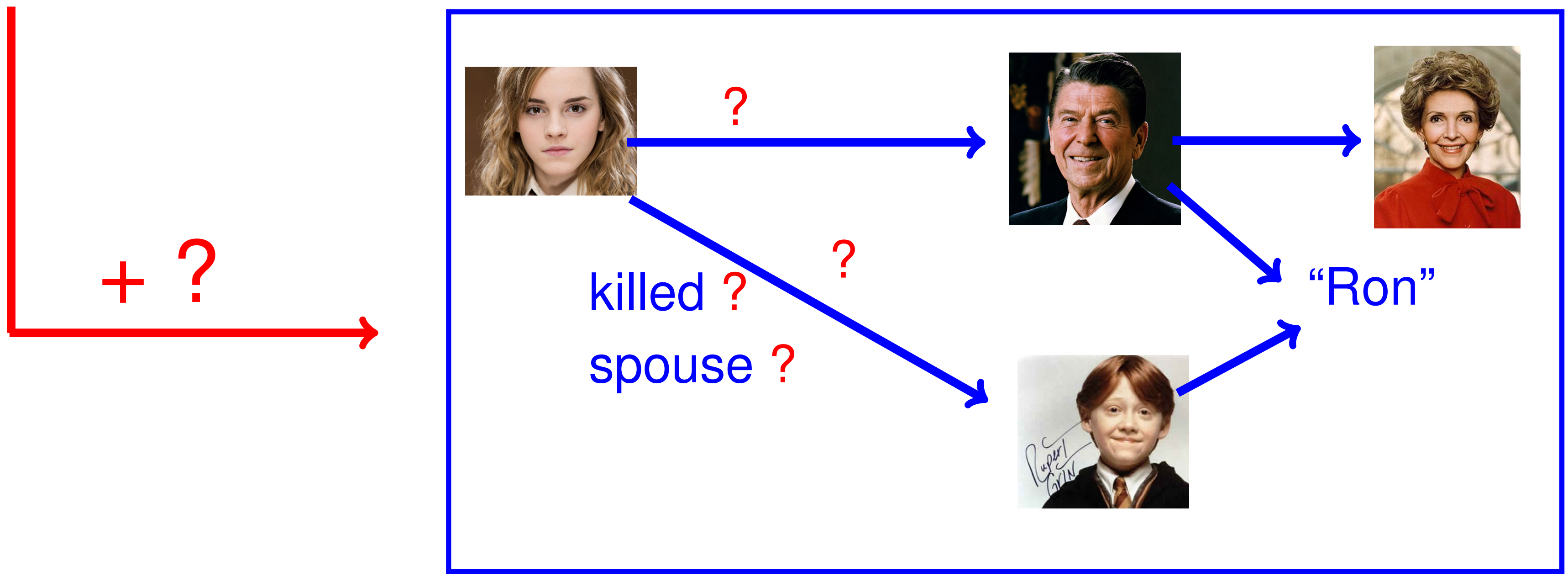
Problems in extending a KB

“Hermione is married to Ron”



Problems in extending a KB

“Hermione is married to Ron”

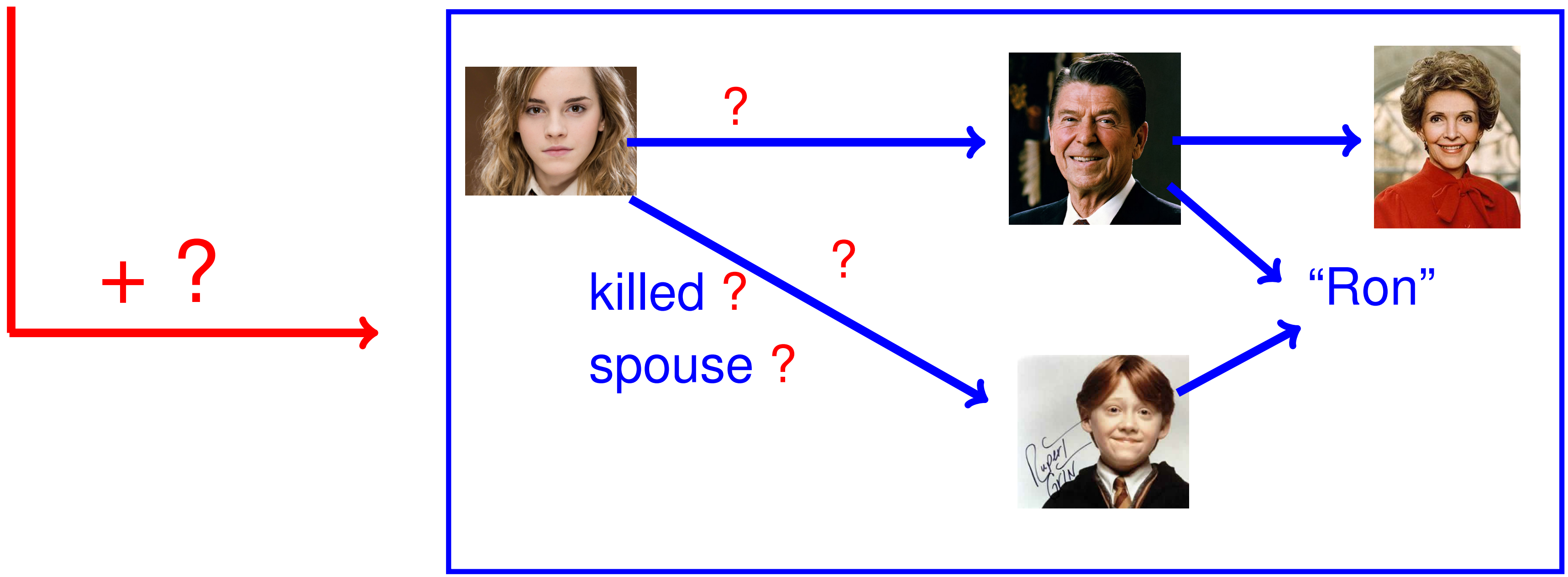


IE faces at least 3 problems:

- Understand patterns (“X is married to Y” = killed(X,Y)?)
- Disambiguate entities (“Ron”= Ronald Reagan?)
- Resolve inconsistencies (Reagan married to 2 women?)

Problems in extending a KB

“Hermione is married to Ron”



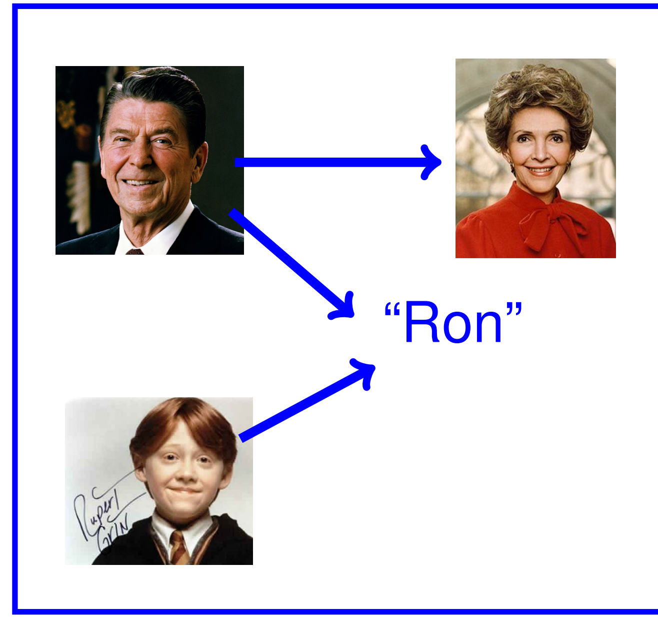
- Disambiguation avoids inconsistency
- Pattern helps disambiguation
- Consistency helps finding pattern

=> Solve all 3 problems together!

Idea: Solve all problems together

“Hermione is married to Ron”

transform everything to logical formulas



$A \wedge B \Rightarrow C$

Find best conclusion

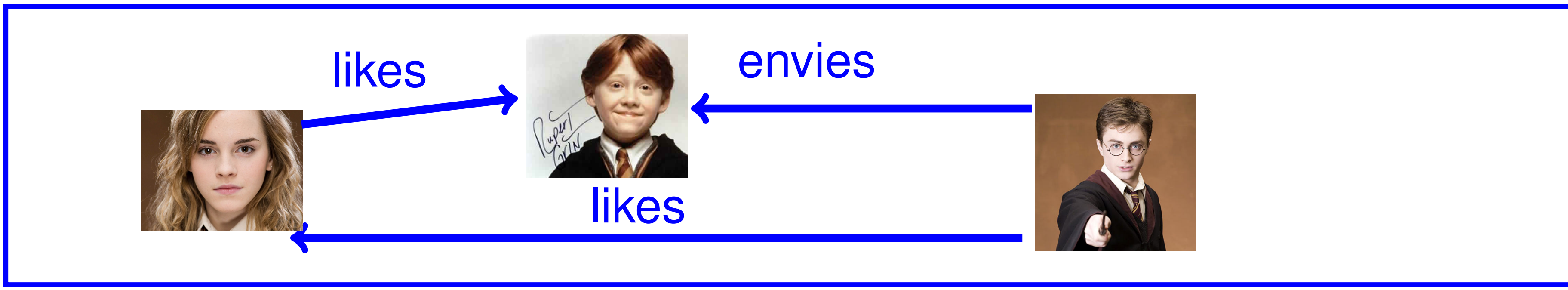
$hasSpouse(Hermione, RonWeasley)$

Refresh: Atoms and KBs

An atom holds (“is true”) in a KB, if it appears in the KB.

A negated atom $\neg A$ holds in a KB if A does not hold.

A conjunction $A \wedge B \wedge \ldots \wedge Z$ holds in a KB, if all of its elements hold.



$likes(Hermione, Ron)$?

$\neg envies(Ron, Harry)$?

$\neg envies(Ron, Harry) \wedge likes(Harry, Hermione)$

$envies(Harry, Ron) \wedge likes(Hermione, Ron) \wedge likes(Harry, Elvis)$

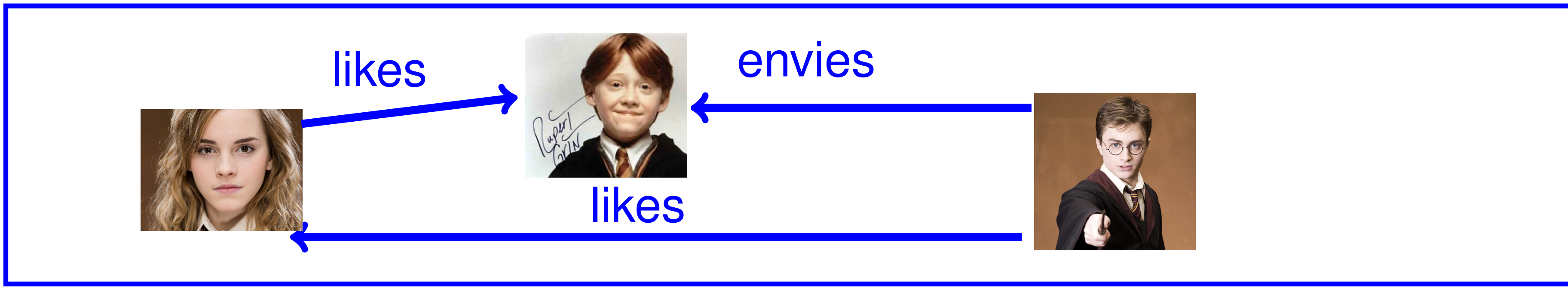
Refresh: Implications

An atom holds ("is true") in a KB, if it appears in the KB.

A negated atom $\neg A$ holds in a KB if A does not hold.

A conjunction $A \wedge B \wedge ... \wedge Z$ holds in a KB, if all of its elements hold.

An implication $\vec{B} \Rightarrow H$ holds in a KB if $\vec{B}$ does not hold or H holds.



$$likes(Hermione, Ron) \Rightarrow likes(Harry, Ron)$$

$$likes(Harry, Ron) \Rightarrow hasSpouse(Harry, Hermione)$$

$$\neg likes(Hermione, Harry) \Rightarrow envies(Harry, Ron)$$

$$\neg likes(Hermione, Harry) \Rightarrow hasSpouse(Harry, Ron)$$

$$likes(Herm., Ron) \wedge \neg envies(Harry, Ron) \Rightarrow ffe(Harry, Ron)$$

Refresh:Universally quantified formulas

A universally quantified formula holds in a KB, if all of its instantiations hold.

$$hasSpouse(x, y)$$

$$hasSpouse(x, y) \Rightarrow hasSpouse(y, x)$$

$$hasSpouse(Elvis, y) \Rightarrow hasSpouse(y, Elvis)$$

$$hasSpouse(Ron, y) \Rightarrow hasSpouse(y, Ron)$$

Def: Weighted Rule

A weighted rule is a rule with an associated real-valued weight.

$$hasSpouse(X, Y) \wedge different(Y, Z) \Rightarrow \neg hasSpouse(X, Z)[3.14]$$

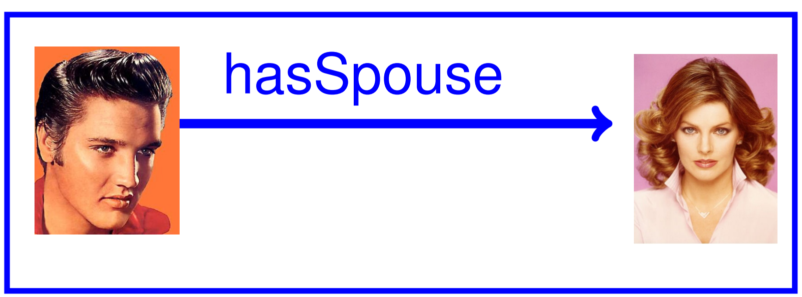
Def: Weight of a KB

Given a KB (a “possible world”) and a set of instantiated rules with weights, the weight of the KB is the sum of the weights of all true rules.

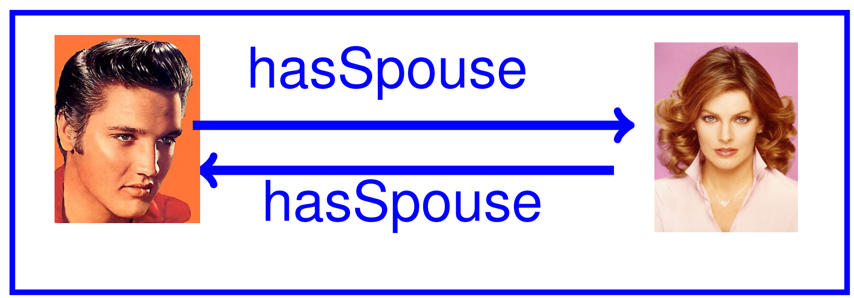
$$hasSpouse(Elvis, Priscilla) \Rightarrow hasSpouse(Priscilla, Elvis)[3]$$

$$hasSpouse(cat, dog) \Rightarrow hasSpouse(dog, cat)[2]$$

KB1



KB2



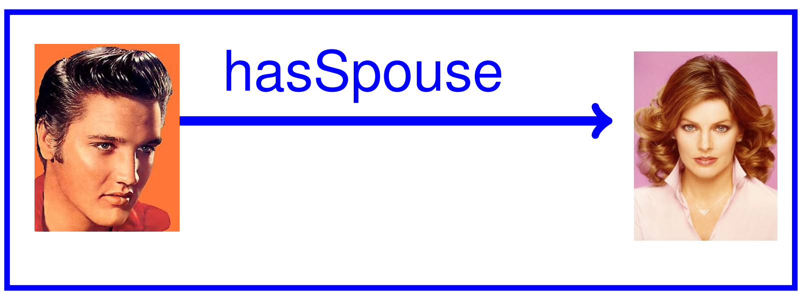
Def: Weight of a KB

Given a KB (a “possible world”) and a set of instantiated rules with weights, the weight of the KB is the sum of the weights of all true rules.

$$hasSpouse(Elvis, Priscilla) \Rightarrow hasSpouse(Priscilla, Elvis)[3]$$

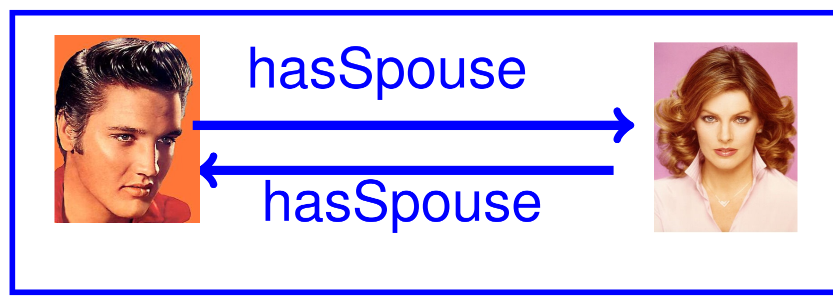
$$hasSpouse(cat, dog) \Rightarrow hasSpouse(dog, cat)[2]$$

KB1



Weight: 2

KB2



Weight: 5

Def: Weighted MAX SAT

Given a set of instantiated rules with weights, weighted MAX SAT is the problem of finding the KB with the highest weight.

(Since SAT is NP-complete, so is MAX SAT and Weighted MAX SAT. There may be multiple such worlds. We are interested in minimal worlds.)

$$is(Ron, immature)[10]$$

$$is(Ron, immature) \wedge type(H., sorceress) \Rightarrow likes(H., Ron)[3]$$

$$type(Hermione, sorceress)[4]$$

Def: Weighted MAX SAT

Given a set of instantiated rules with weights, weighted MAX SAT is the problem of finding the KB with the highest weight.

(Since SAT is NP-complete, so is MAX SAT and Weighted MAX SAT. There may be multiple such worlds. We are interested in minimal worlds.)

$$is(Ron, immature)[10]$$

$$is(Ron, immature) \wedge type(H., sorceress) \Rightarrow likes(H., Ron)[3]$$

$$type(Hermione, sorceress)[4]$$

Best world:

$$is(Ron, immature)$$
$$type(Hermione, sorceress)$$
$$likes(Hermione, Ron)$$

weight: 17

Task: Weighted MAX SAT

Find the KB with the highest weight:

$is(Hermione, smart)$[1]

$is(Herm., smart) \wedge is(Harry, smart) \Rightarrow likes(Herm., Harry)$[3]

$likes(Hermione, Ron) \Rightarrow \neg likes(Hermione, Harry)$[100]

$is(Harry, smart)$[10]

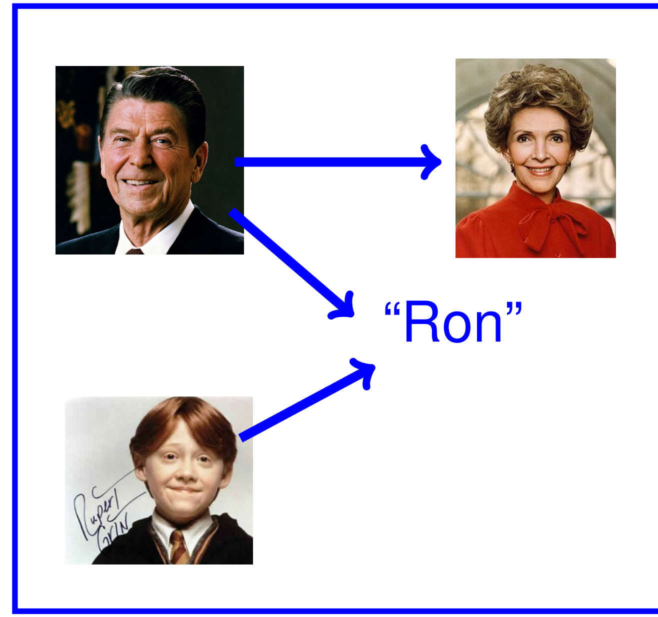
$likes(Hermione, Ron)$[20]

Hint: Start by satisfying disjunctions with one literal and high weight.

Back to our problem

“Hermione is married to Ron”

transform everything to logical formulas



$$A \wedge B \Rightarrow C$$

Find best conclusion

$$hasSpouse(Hermione, RonWeasley)$$

Consistency

Consistency constraints can be expressed by rules:

$$hasSpouse(X,Y) \wedge different(Y,Z) \Rightarrow \neg hasSpouse(X,Z)\text{[10]}$$

$$hasSpouse(X,Y) \Rightarrow type(X,person)\text{[20]}$$

$$loves(X,Y) \wedge \neg hasSpouse(X,Y) \Rightarrow type(X,stupidPerson)\text{[3]}$$

...

Rules and weights can be designed manually. Such rules will guide our information extraction process.

A KB can be expressed as rules

Every fact from the KB can be expressed as a weighted rule:

$$type(Hermione, Person)[100]$$

High weight

This is corresponds to the rule

$$\Rightarrow type(Hermione, Person)[100]$$

Assuming completeness

If we assume the KB to be perfect on a relation, we can create negative rules:

$\neg type(Hermione, X)$[100]

$\forall X : type(Hermione, X) \notin KB$

i.e.

$\neg type(Hermione, chicken)$[100]

$\neg type(Hermione, cat)$[100]

$\neg type(Hermione, mouse)$[100]

...

Expressing the corpus as rules

D42: Hermione married Ron.

$occurs$(Hermione@D42, “married”, Ron@D42)

Expressing the corpus as rules

D42: Hermione married Ron.

$occurs$(Hermione@D42, “married”, Ron@D42)

Does not talk about Ronald Reagan or Ron Weasley, but about the word “Ron” in document D42.

Ron@D42 is a “word in context” (wic).

Expressing the corpus as rules

D42: Hermione married Ron.

$occurs$(Hermione@D42, “married”, Ron@D42)

The word “Ron” in document D42 can mean different entities (from KB):

$means$(Ron@D42, $RonaldReagan$)

$means$(RonD42, $RonWeasley$)

But only one entity in practice:

$$means(X, Y) \wedge different(Y, Z) \Rightarrow \neg means(X, Z)$$

Weights for corpus rules

$occurs$(Hermione@D42, “married”, Ron@D42)[3]

$means$(Ron@D42, $RonaldReagan$)

$means$(Ron@D42, $RonWeasley$)

$$means(X, Y) \wedge different(Y, Z) \Rightarrow \neg means(X, Z)$$

Weights for corpus rules

occurrences

$occurs$(Hermione@D42, “married”, Ron@D42)[3]

$means$(Ron@D42, $RonaldReagan$)

$means$(Ron@D42, $RonWeasley$)

$$means(X, Y) \wedge different(Y, Z) \Rightarrow \neg means(X, Z)$$

Weights for corpus rules

$occurs$(Hermione@D42, “married”, Ron@D42)[3]

From disambiguation by context/prior

$means$(Ron@D42, $RonaldReagan$)[5]

$means$(Ron@D42, $RonWeasley$)[7]

$$means(X, Y) \wedge different(Y, Z) \Rightarrow \neg means(X, Z)$$

Weights for corpus rules

$occurs$(Hermione@D42, “married”, Ron@D42)[3]

$means$(Ron@D42, $Ronald Reagan$)[5]

$means$(Ron@D42, $Ron Weasley$)[7]

“Hard” rule with very high weight

$means(X, Y) \wedge different(Y, Z) \Rightarrow \neg means(X, Z)$ [100]

Weights for corpus rules

$occurs$(Hermione@D42, “married”, Ron@D42)[3]

$means$(Ron@D42, $RonaldReagan$)

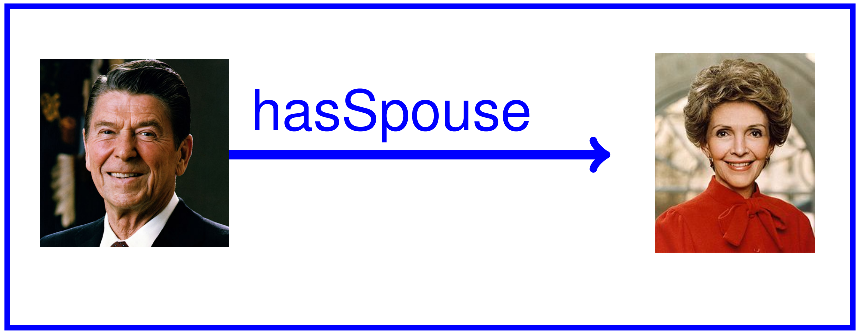
$means$(Ron@D42, $RonWeasley$)

$means(X,Y) \wedge different(Y,Z) \Rightarrow \neg means(X,Z)$ [100]

Let us ignore the weights for a moment.

Deducing patterns

KB



+

Reagan married Davis.

=

"X married Y"

is pattern for

$hasSpouse$(X,Y)

Deducing patterns

KB

hasSpouse

$hasSpouse(Reagan, Davis)$

+

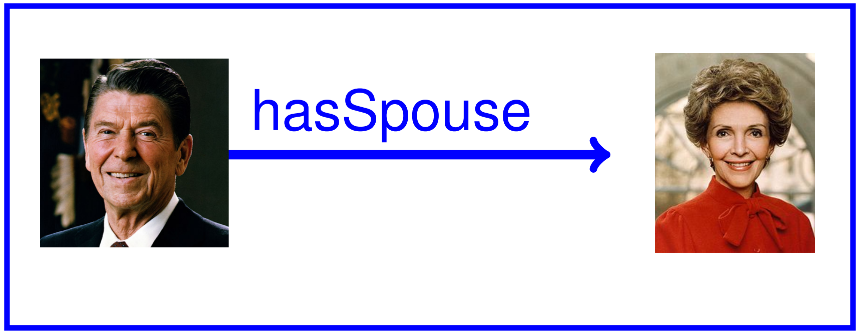
Reagan married Davis.

=

"X married Y"
is pattern for
$hasSpouse$(X,Y)

Deducing patterns

KB



+

Reagan married Davis.

=

"X married Y"
is pattern for
$hasSpouse$(X,Y)

$hasSpouse(Reagan, Davis)$

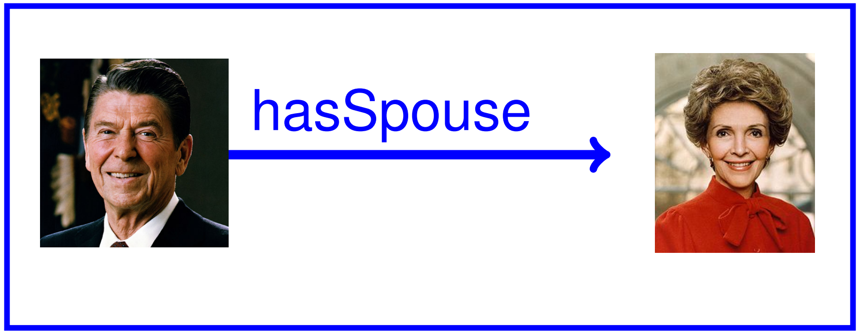
$occurs$(R@1,"married", D@1)

$means$(R@1, $Reagan$)

$means$(D@1, $Davis$)

Deducing patterns

KB



+

Reagan married Davis.

=

“X married Y”
is pattern for
$hasSpouse$(X,Y)

$hasSpouse(Reagan, Davis)$

$occurs$(R@1,“married”, D@1)

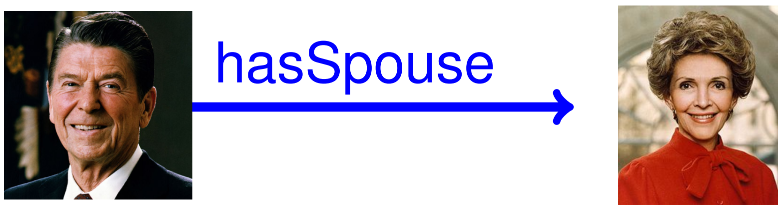
$means$(R@1, $Reagan$)

$means$(D@1, $Davis$)

$$occurs(X, P, Y)$$
$$\wedge means(X, X')$$
$$\wedge means(Y, Y')$$
$$\wedge R(X', Y')$$
$$\Rightarrow isPatternFor(P, R)$$

Deducing patterns

KB



+

Reagan married Davis.

=

"X married Y"
is pattern for
$hasSpouse$(X,Y)

$hasSpouse(Reagan, Davis)$

$occurs$(R@1,"married", D@1)

$means$(R@1, $Reagan$)

$means$(D@1, $Davis$)

$$occurs(X, P, Y) \wedge means(X, X') \wedge means(Y, Y') \wedge R(X', Y') \Rightarrow isPatternFor(P, R)$$

$isPatternFor$("married", $hasSpouse$)

Applying patterns

"X married Y"

is pattern for

$hasSpouse$(X,Y)

+

Elvis married Priscilla.

=

Applying patterns

“X married Y”
is pattern for
$hasSpouse$(X,Y)

$isPatternFor$(“married”,$hasSpouse$)

+

Elvis married Priscilla.

=

Applying patterns

"X married Y"
is pattern for
$hasSpouse$(X,Y)

+

Elvis married Priscilla.

=

$isPatternFor$("married",$hasSpouse$)

$occurs$(E@1, "married", P@1)

$means$(E@1, $Elvis$)

$means$(P@1, $Priscilla$)

Applying patterns

"X married Y"
is pattern for
$hasSpouse$(X,Y)

+

Elvis married Priscilla.

=

$isPatternFor$("married",$hasSpouse$)

$occurs$(E@1, "married", P@1)

$means$(E@1, $Elvis$)

$means$(P@1, $Priscilla$)

$$
\begin{aligned}
& occurs(X, P, Y) \\
& \quad \wedge means(X, X') \\
& \quad \wedge means(Y, Y') \\
& \quad \wedge isPatternFor(P, R) \\
& \quad \Rightarrow R(X', Y')
\end{aligned}
$$

Applying patterns

"X married Y"
is pattern for
$hasSpouse$(X,Y)

$isPatternFor$("married",$hasSpouse$)

+

$occurs$(E@1, "married", P@1)

$means$(E@1, $Elvis$)

Elvis married Priscilla.

$means$(P@1, $Priscilla$)

=

$$
\begin{aligned}
& occurs(X, P, Y) \\
& \quad \wedge means(X, X') \\
& \quad \wedge means(Y, Y') \\
& \quad \wedge isPatternFor(P, R) \\
& \quad \Rightarrow R(X', Y')
\end{aligned}
$$

$hasSpouse(Elvis, Priscilla)$

task>106

Task: Pattern deduction by rules

Pattern deduction:

$$occurs(X, P, Y) \\ \wedge means(X, X') \\ \wedge means(Y, Y') \\ \wedge R(X', Y') \\ \Rightarrow isPatternFor(P, R)$$

Pattern application:

$$occurs(X, P, Y) \\ \wedge means(X, X') \\ \wedge means(Y, Y') \\ \wedge isPatternFor(P, R) \\ \Rightarrow R(X', Y')$$

1 : $occurs$(P@1, “adores”, E@1)

2 : $means$(E@1,$Elvis$)

3 : $means$(P@1, $Priscilla$)

4 : $hasSpouse(Priscilla, Elvis)$

5 : $occurs$(M@1, “adores”, E@1)

6 : $means$(M@1, $Madonna$)

All rules have weight 1.

Compute facts that will be in the best world.

Life is not easy

- words are ambiguous

  “Ron”

- corpora may err

  “Madonna is married to Elvis”

  $occurs$(W@1, “eats”, R@1)

- parsing may fail

  “Coyote dreams Coyote eats Roadrunner”

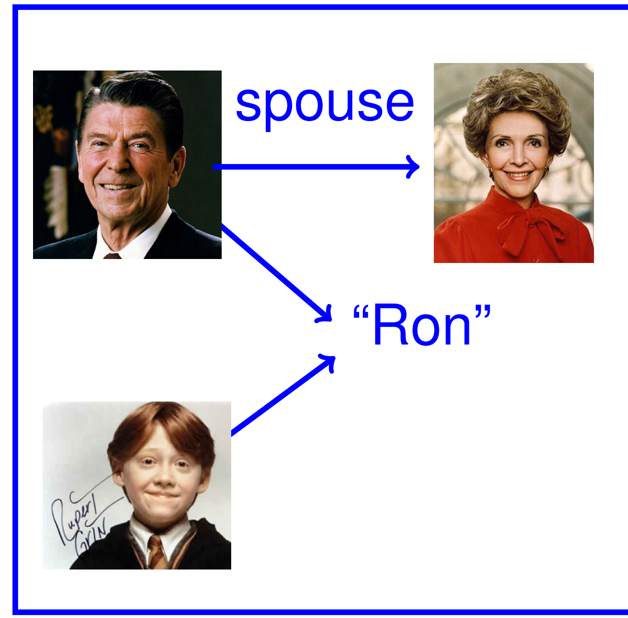
- contradictions may occur

  Reagan was married twice.

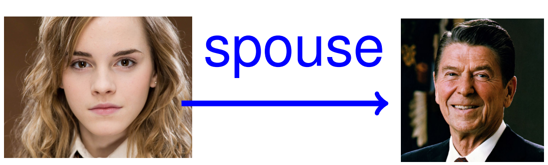
=> we will compute the most plausible world

Finding the most plausible world

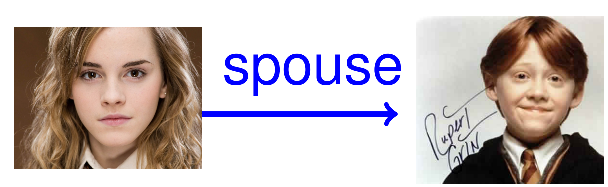
“Hermione married Ron”



World1:



World2:

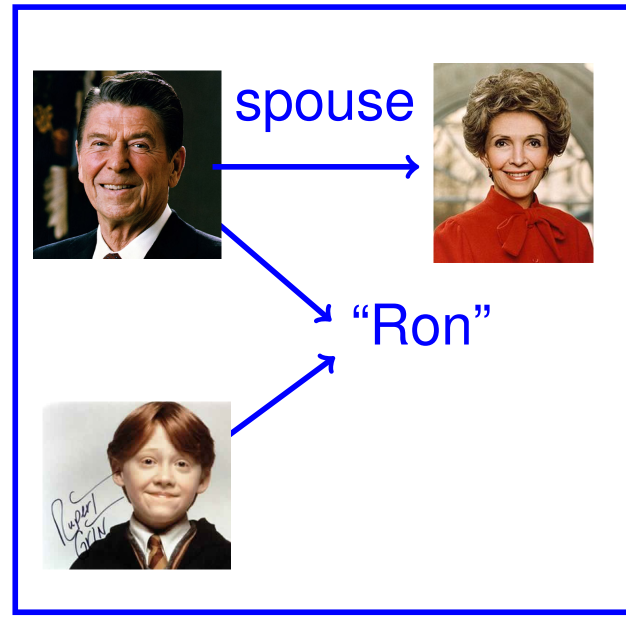


Finding the most plausible world

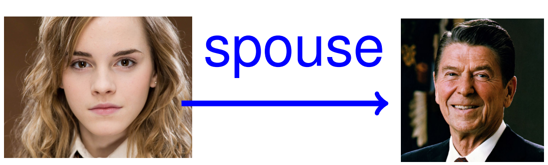
“Hermione married Ron”

$occurs$(H@1, “married”,R@ 1)[1]

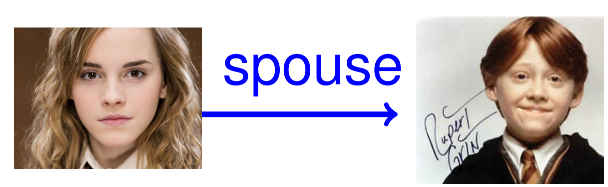
$isPatternFor$(“married”, $spouse$)[1]



World1:



World2:



Finding the most plausible world

"Hermione married Ron"

$occurs$(H@1, "married",R@ 1)[1]

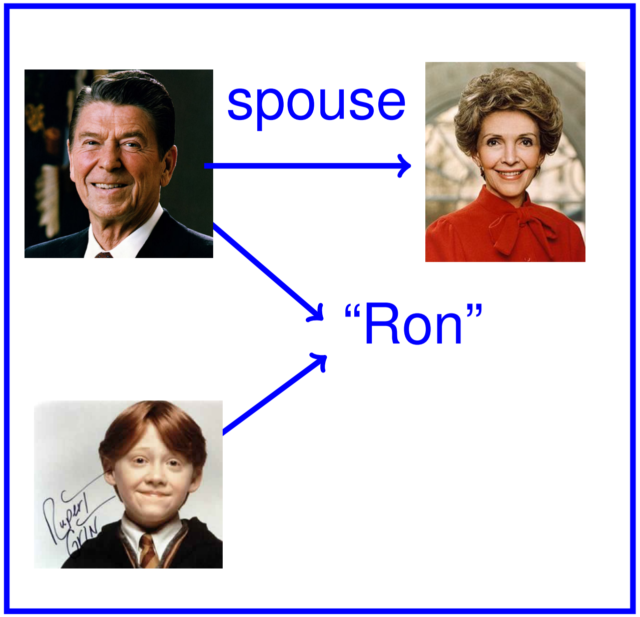
$isPatternFor$("married", $spouse$)[1]

$means$(H@1, $Hermione$)[5]

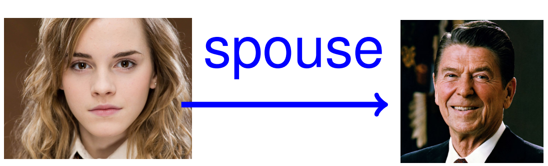
$means$(R@1, $RonWeasley$)[2]

$means$(R@1, $Reagan$)[3]

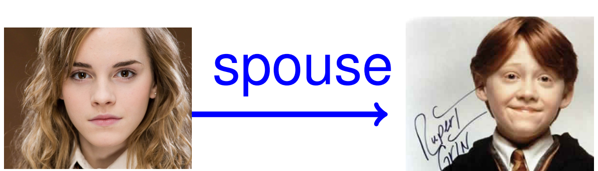
$means(X,Y) \wedge Y \neq Z \Rightarrow \neg means(X,Z)$[10]



World1:



World2:



Finding the most plausible world

“Hermione married Ron”

$occurs$(H@1, “married”,R@ 1)[1]

$isPatternFor$(“married”, $spouse$)[1]

$means$(H@1, $Hermione$)[5]

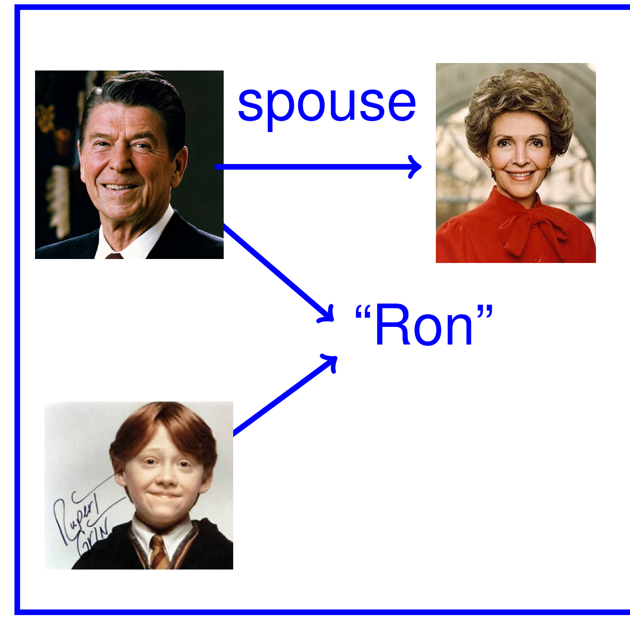
$means$(R@1, $RonWeasley$)[2]

$means$(R@1, $Reagan$)[3]

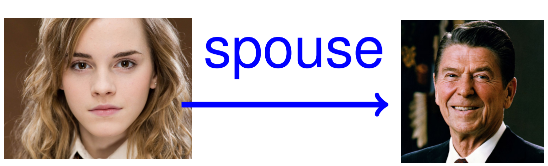
$means(X,Y) \wedge Y \neq Z \Rightarrow \neg means(X,Z)$[10]

$spouse(X,Y) \wedge Y \neq Z \Rightarrow \neg spouse(Z,X)$[6]

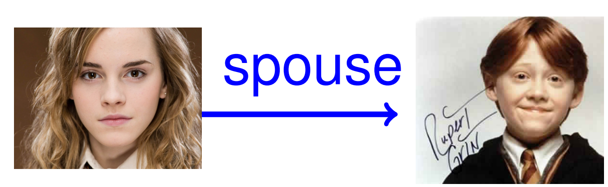
\+ Pattern Application Rule [10]



World1:



World2:



Finding the most plausible world

"Hermione married Ron"

$occurs$(H@1, "married",R@ 1)[1]

$isPatternFor$("married", $spouse$)[1]

$means$(H@1, $Hermione$)[5]

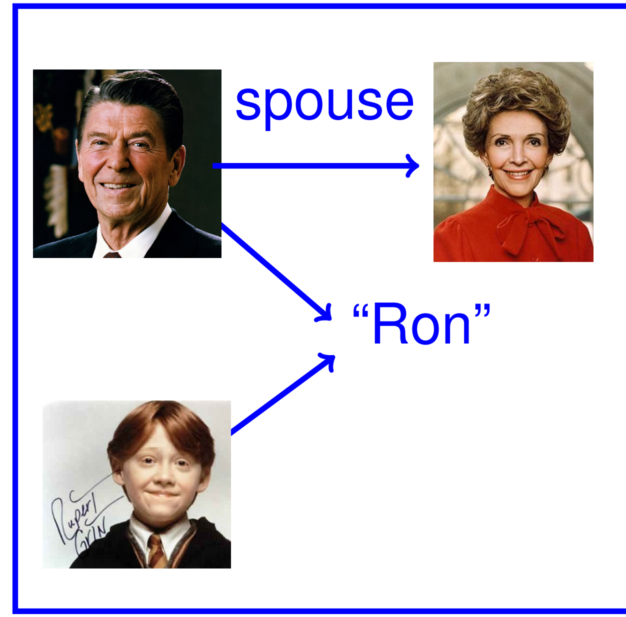
$means$(R@1, $RonWeasley$)[2]

$means$(R@1, $Reagan$)[3]

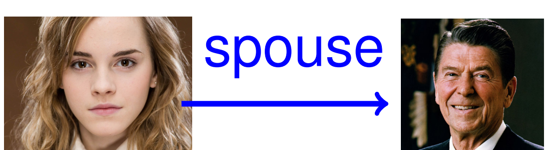
$means(X, Y) \wedge Y \neq Z \Rightarrow \neg means(X, Z)$[10]

$spouse(X, Y) \wedge Y \neq Z \Rightarrow \neg spouse(Z, X)$[6]

+ Pattern Application Rule [10]



World1:

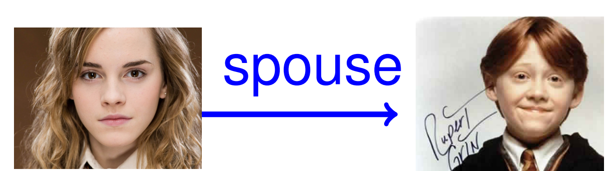


loses 2

wins 3

loses 6

World2:



Finding the most plausible world

"Hermione married Ron"

$occurs$(H@1, "married",R@ 1)[1]

$isPatternFor$("married", $spouse$)[1]

$means$(H@1, $Hermione$)[5]

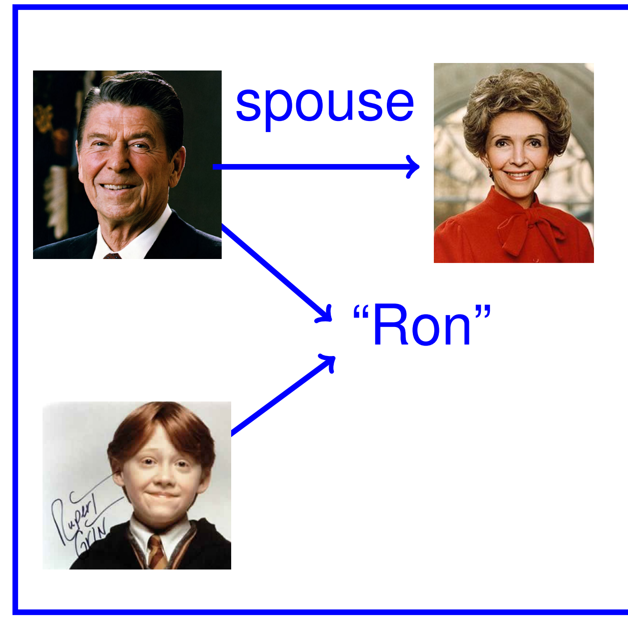
$means$(R@1, $RonWeasley$)[2]

$means$(R@1, $Reagan$)[3]

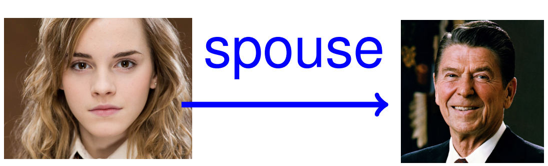
$means(X,Y) \wedge Y \neq Z \Rightarrow \neg means(X,Z)$[10]

$spouse(X,Y) \wedge Y \neq Z \Rightarrow \neg spouse(Z,X)$[6]

+ Pattern Application Rule [10]



World1:

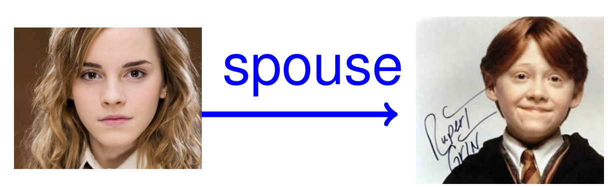


loses 2

wins 3

loses 6

World2:



loses 3

wins 6

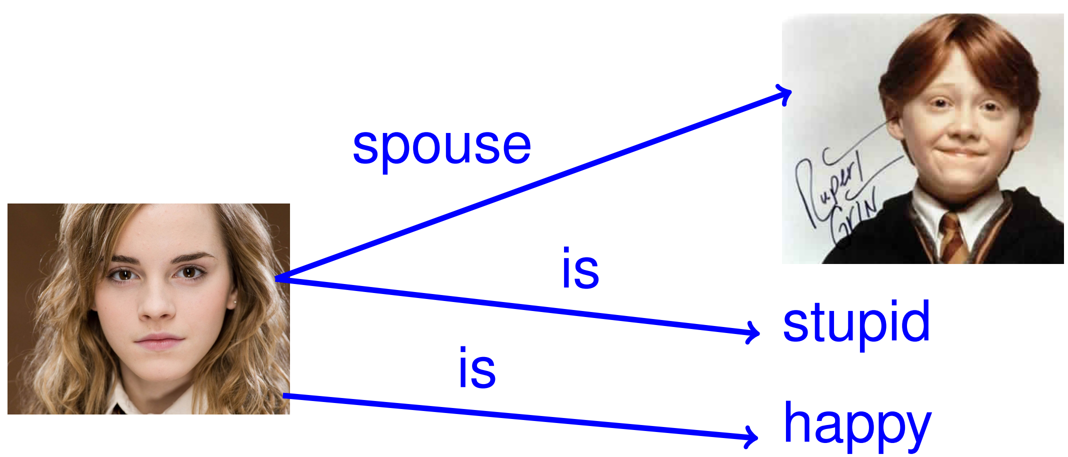
wins 2

Summary: IE by reasoning

IE by reasoning converts

- background knowledge (existing KB)
- logical constraints
- the corpus

into logical rules and aims to find the most plausible possible world given these evidences.



References

SOFIE - a self-organizing framework for IE

->markov-logic

->semantic-web