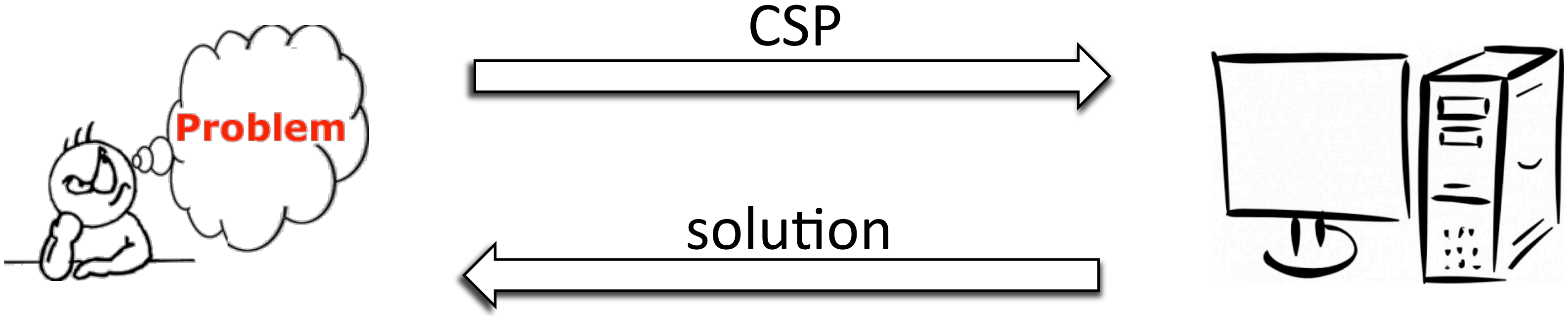
Constraint Acquisition via Partial Queries

Christian Bessiere[1] Remi Coletta[1] Emmanuel Hebrard[2] George Katsirelos[3]
Nadjib Lazaar[1] Nina Narodytska[4] Claude-Guy Quimper[5] Toby Walsh[4]

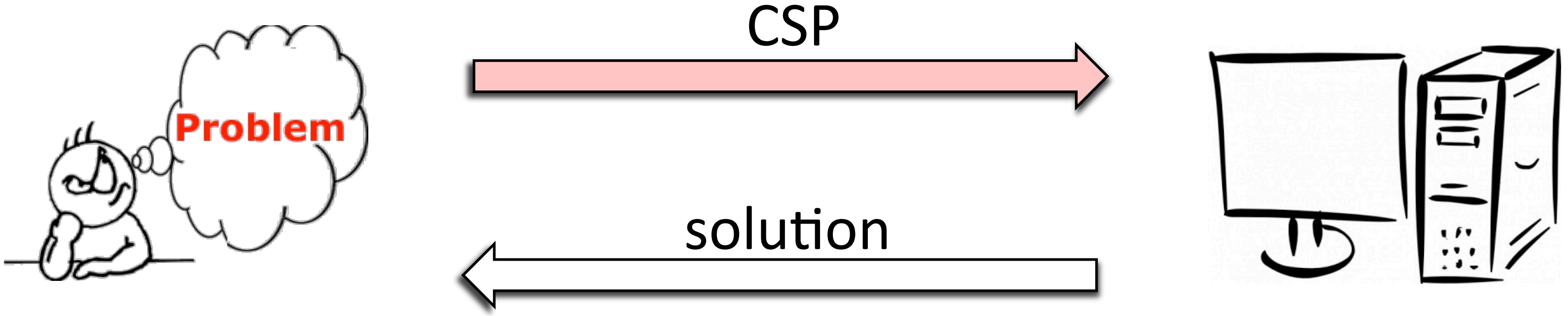
[1]CNRS, U. Montpellier, France
[2]LAAS-CNRS, Toulouse, France
[3]INRA Toulouse, France
[4]NICTA, UNSW, Sydney, Australia
[5]U. Laval, Quebec City, Canada

jfpc 2014

Motivations

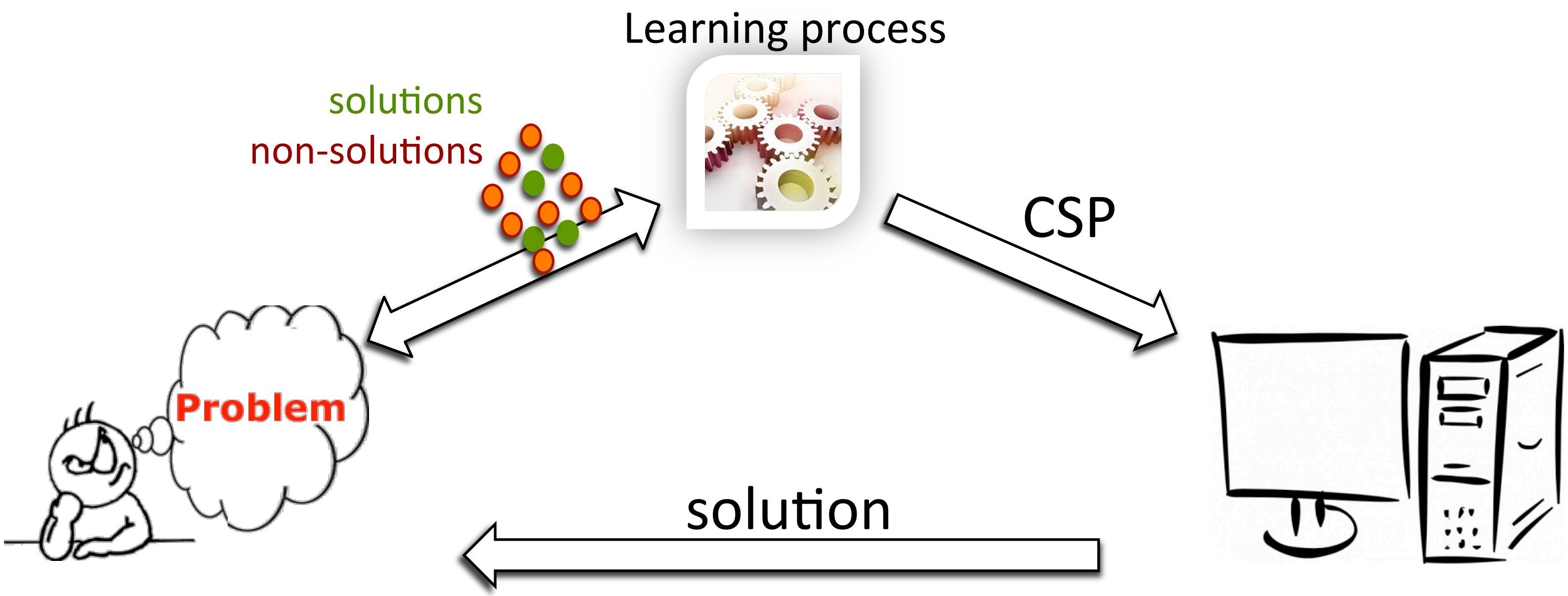


Motivations



- Question: How does the user write down the constraints of a problem?
- Limitations: modelling constraint networks require a fair expertise
- Need: Simple way to build constraint model ➔ Modeller-assistant

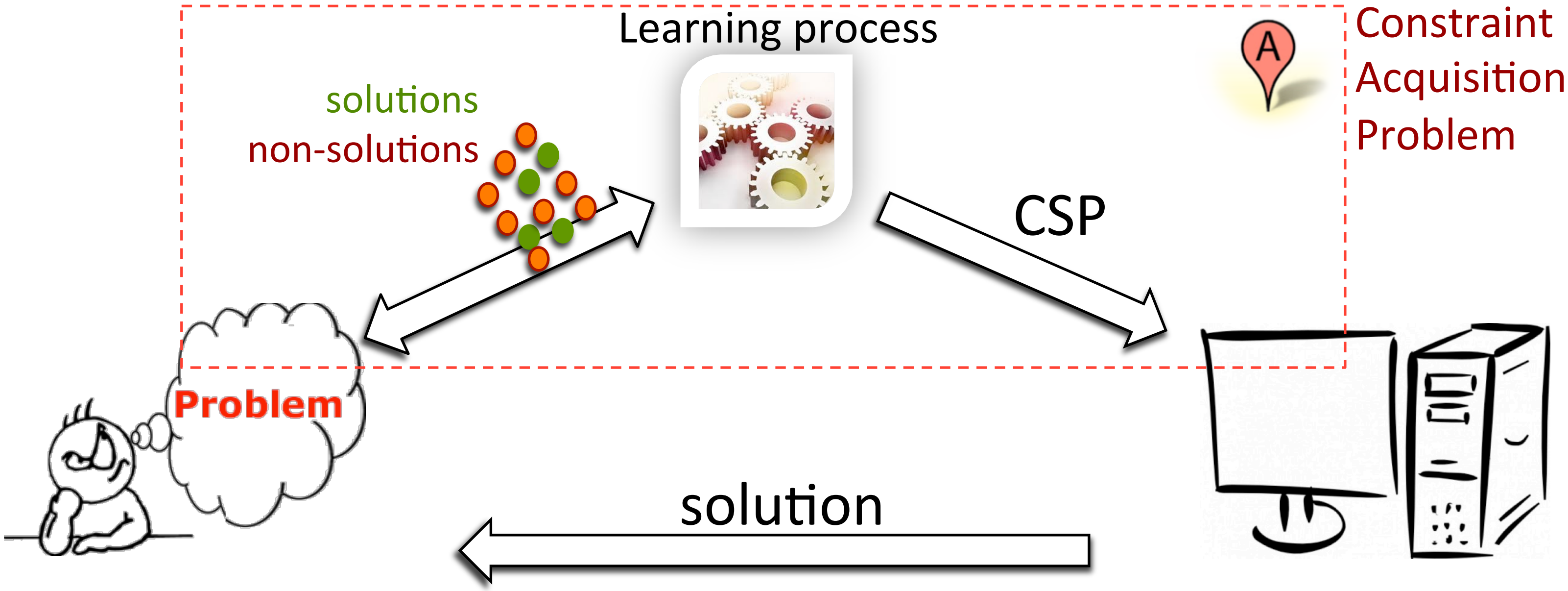
Motivations



- Question: How does the user write down the constraints of a problem?
- Limitations: modelling constraint networks require a fair expertise

- Need: Simple way to build constraint model ➔ Modeller-assistant
- How: In a Machine Learning way (passive/active, offline/online, by reinforcement…)

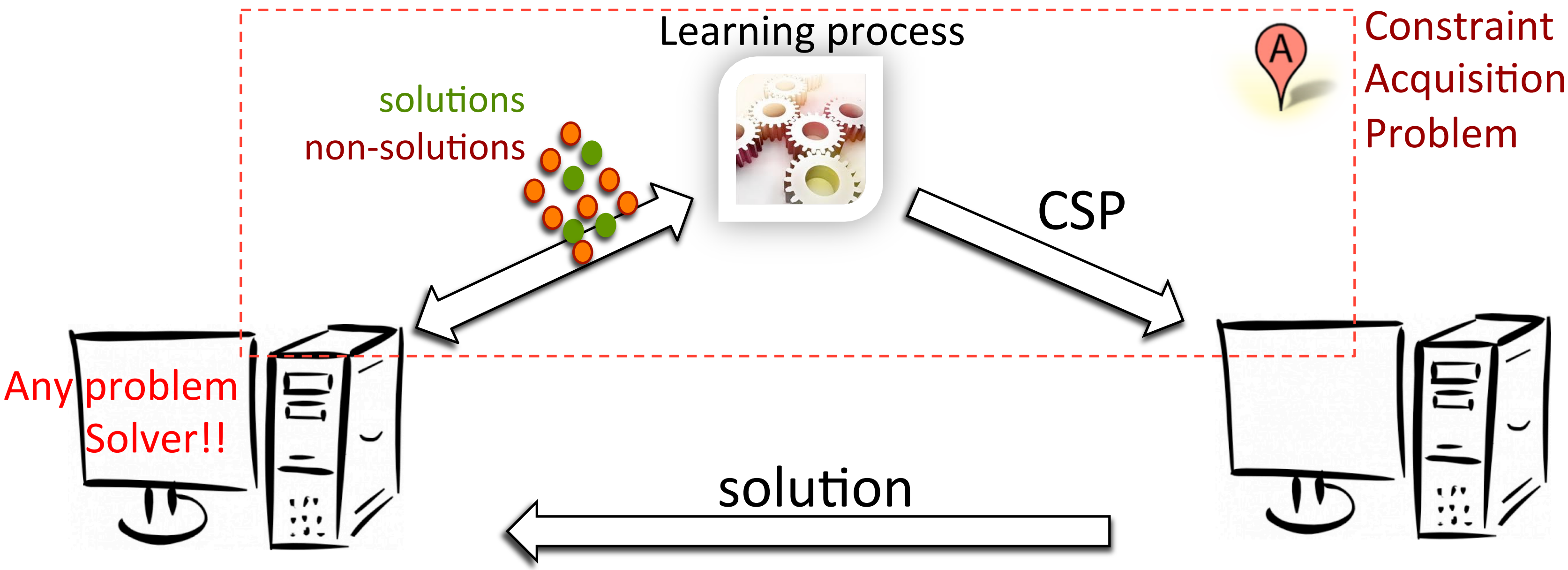
Motivations



- Question: How does the user write down the constraints of a problem?
- Limitations: modelling constraint networks require a fair expertise

- Need: Simple way to build constraint model ➔ Modeller-assistant
- How: In a Machine Learning way (passive/active, offline/online, by reinforcement…)

Motivations

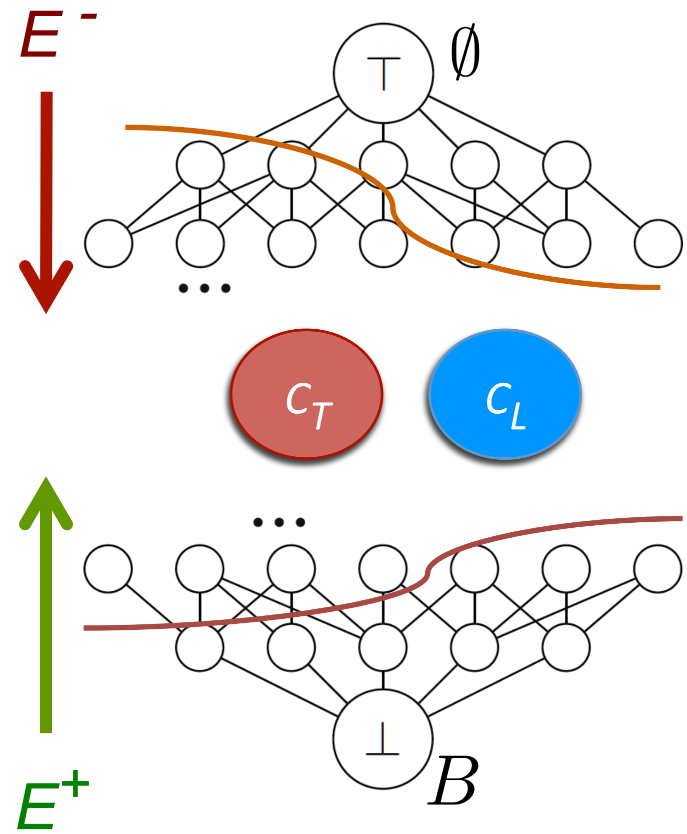


- Question: How does the user write down the constraints of a problem?
- Limitations: modelling constraint networks require a fair expertise

- Need: Simple way to build constraint model ➔ Modeller-assistant
- How: In a Machine Learning way (passive/active, offline/online, by reinforcement…)

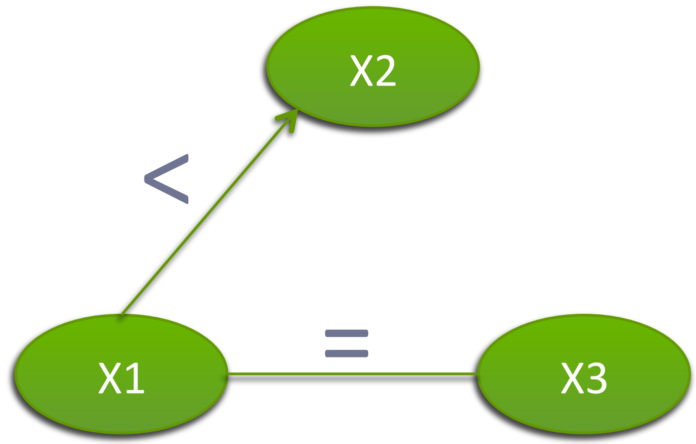
Constraint Acquisition Problem

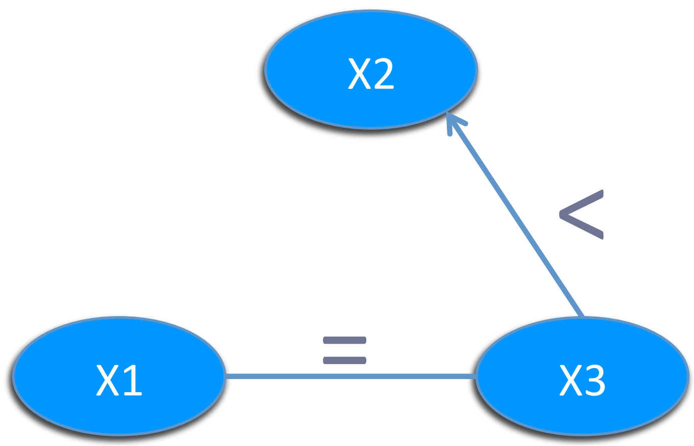
- **Inputs:**
  - (X, D) : Vocabulary
  - B : Bias (possible constraints)
  - C_T : Target network
  - (E^+, E^-) : positives and negatives
- **Output:**
  - C_L: Learnt network s.t.,
    - $C_L \subset B : C_L \equiv C_T$



Example

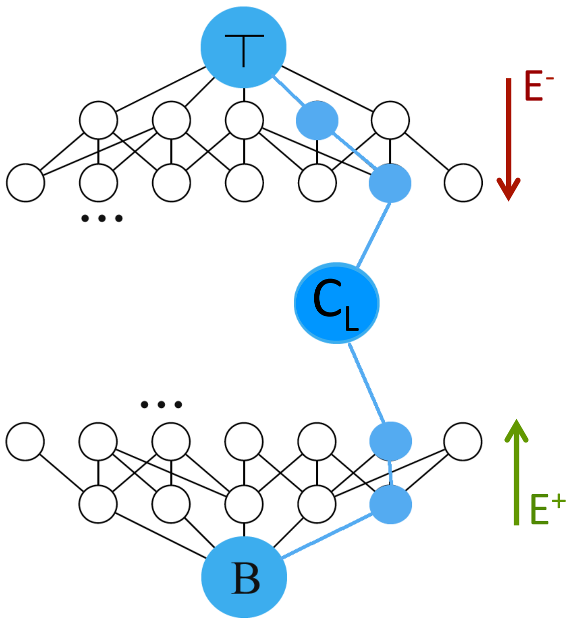
- $\Gamma = \{<, =\}$
- $B = \{x_i < x_j, x_i = x_j, \forall i, j\}$
- $C_T = \{x_1 = x_3, x_1 < x_2\}$
- $C_L = \{x_1 = x_3, x_3 < x_2\}$





State of the art

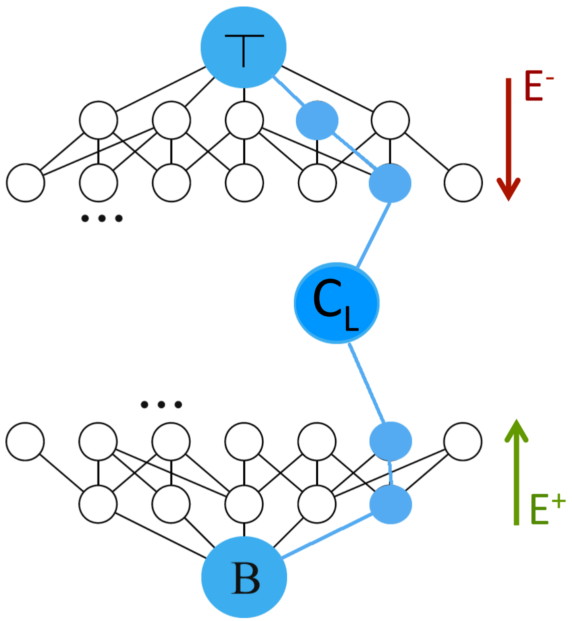
- CONACQ
  - SAT-Based constraint acquisition
  - Bidirectional search
  - Conacq1.0 (passive learning) [Bessiere et al. ECML05]
  - Conacq2.0 (active learning) [Bessiere et al. IJCAI07]



State of the art

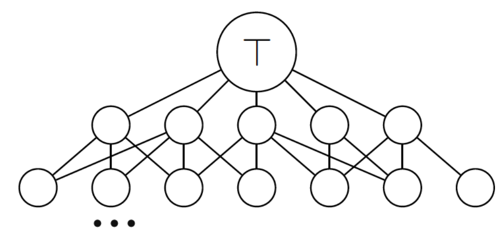
- CONACQ
  - SAT-Based constraint acquisition
  - Bidirectional search
  - Conacq1.0 (passive learning) [Bessiere et al. ECML05]
  - Conacq2.0 (active learning) [Bessiere et al. IJCAI07]

$$\mathcal{K} = \underbrace{(\neg x_1 \wedge \neg x_2 \wedge \neg x_3)}_{e^+} \bigwedge \underbrace{(x_4 \vee x_5 \vee x_6 \vee x_7)}_{e^-} \ldots$$



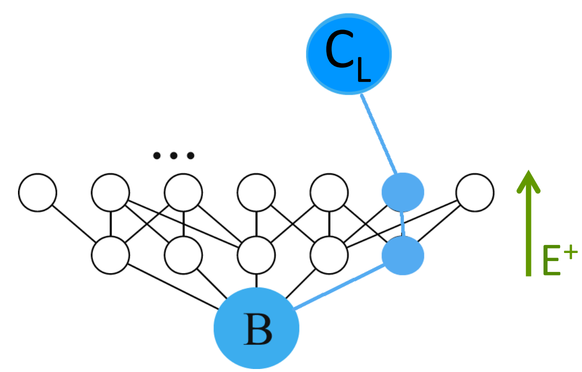
State of the art

- **CONACQ**
  - SAT-Based constraint acquisition
  - Bidirectional search
  - Conacq1.0 (passive learning) [Bessiere et al. ECML05]
  - Conacq2.0 (active learning) [Bessiere et al. IJCAI07]



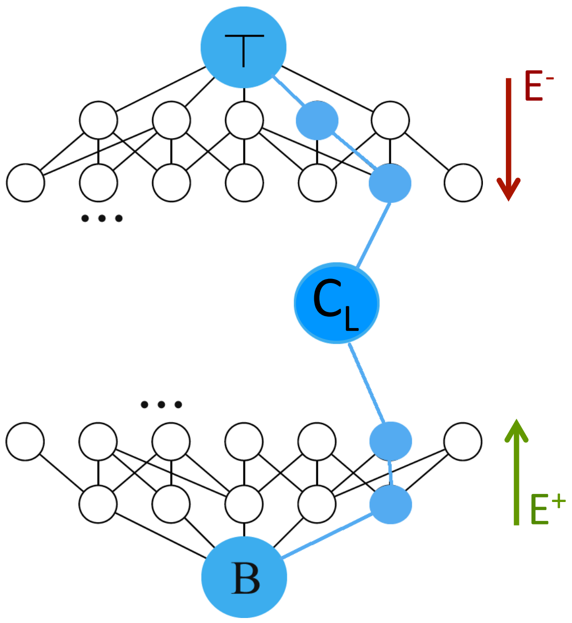
$$\mathcal{K} = \underbrace{(\neg x_1 \wedge \neg x_2 \wedge \neg x_3)}_{e^+} \bigwedge \underbrace{(x_4 \vee x_5 \vee x_6 \vee x_7)}_{e^-} \ldots$$

- **ModelSeeker** [Beldiceanu and Simonis, CP12]
  - A passive learning
  - Based on global constraint catalog (more than 400)
  - Buttom-up search

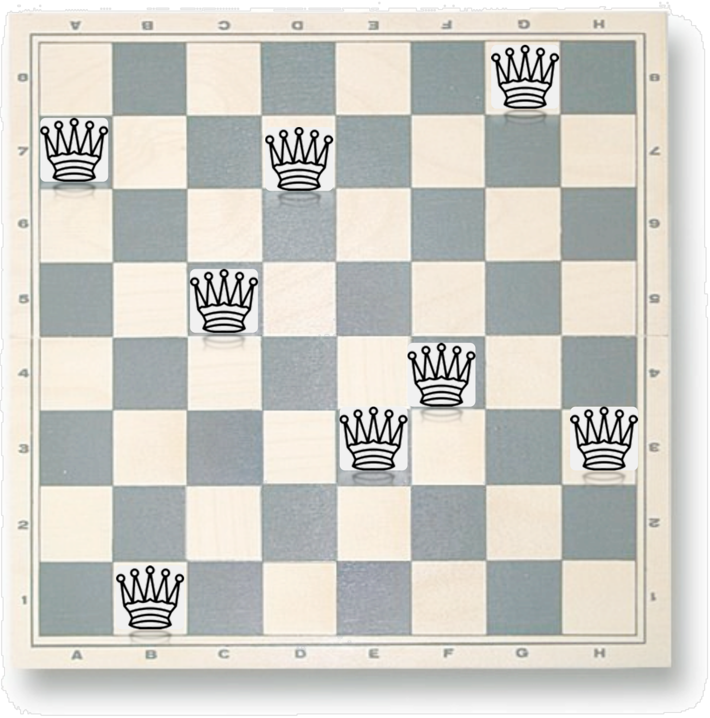


QUACQ: Quick Acquisition

- QUACQ [Bessiere et al. IJCAI13]
  - Active learning approach
  - Bidirectional search
    - But it can be top-down search only if no positive example
  - Based on partial queries to elucidate the scope of the constraint to learn

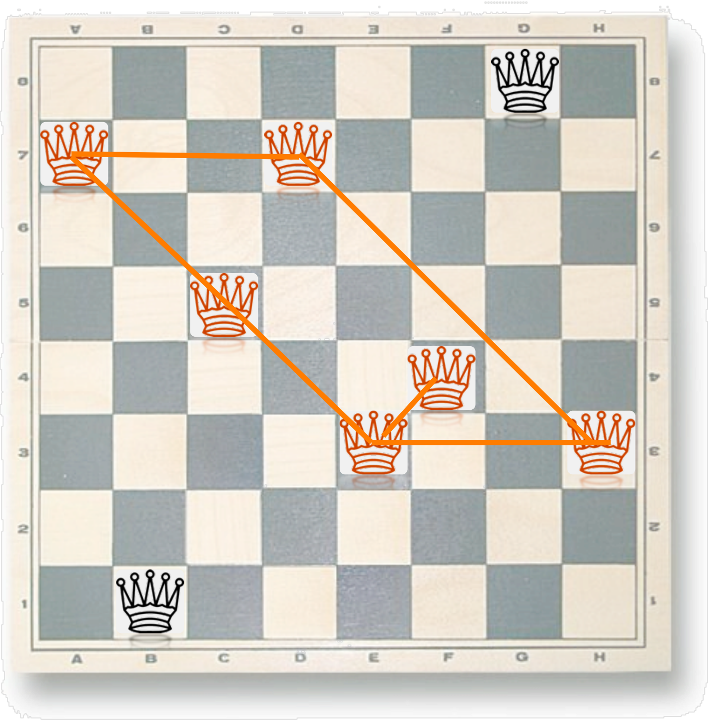


Partial Queries



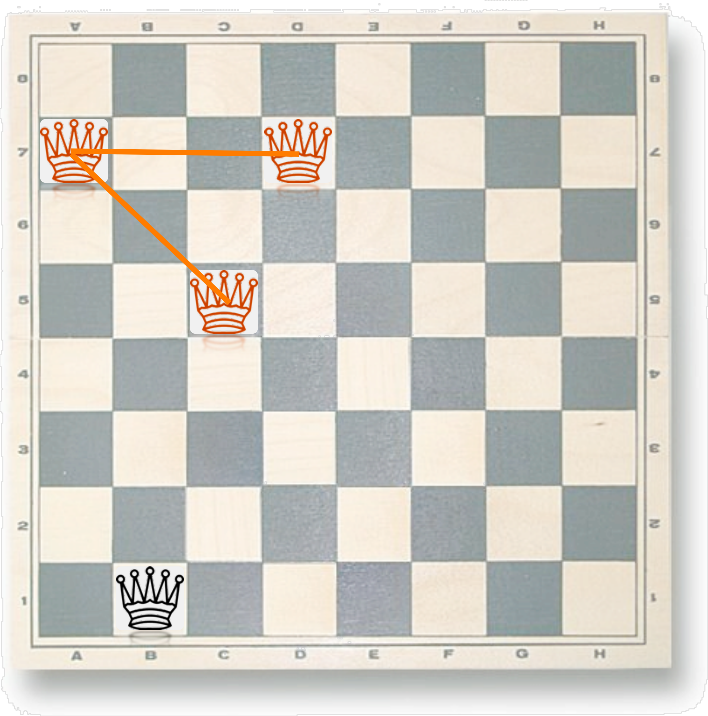
ask(2, 8, 4, 2, 6, 5, 1, 6)

Partial Queries



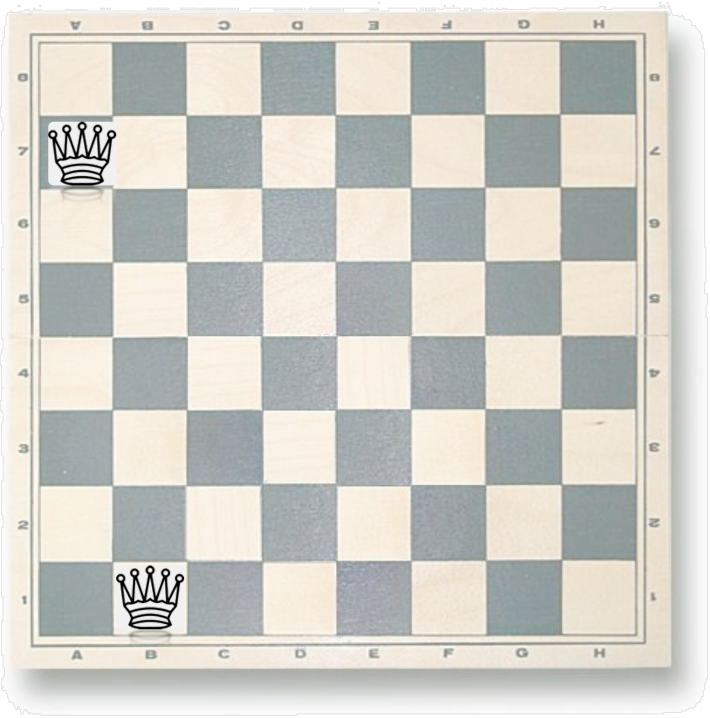
ask(2, 8, 4, 2, 6, 5, 1, 6) = No

Partial Queries



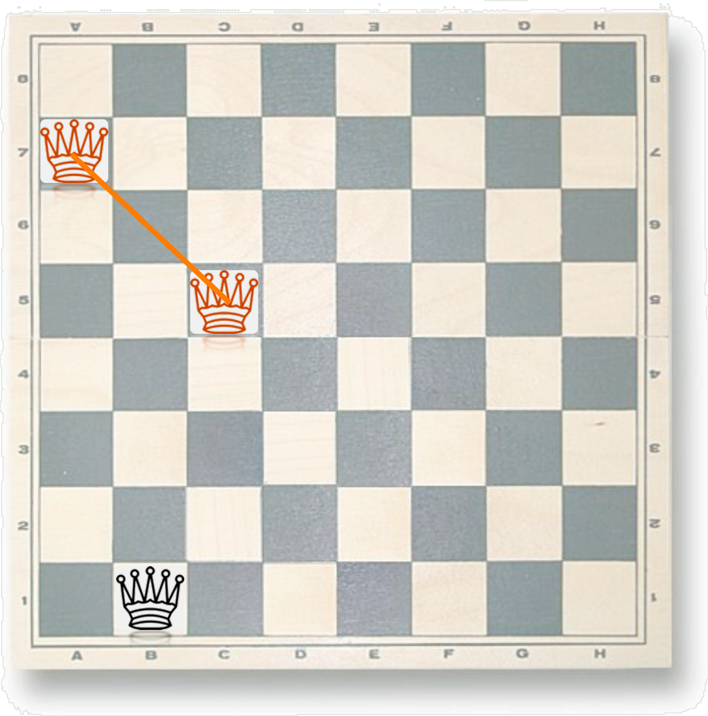
ask(2, 8, 4, 2, -, -, -, -) = No

Partial Queries



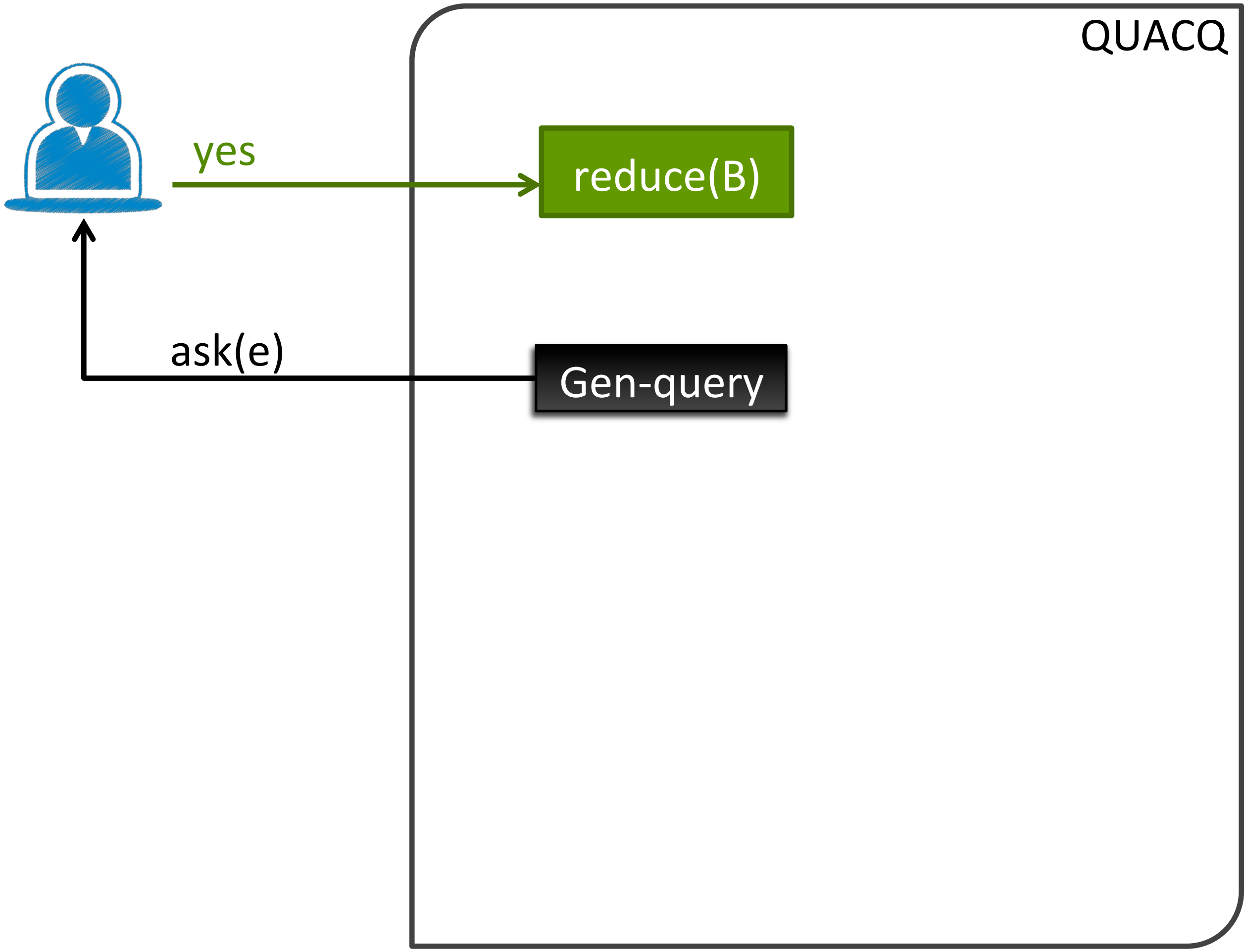
ask(2, 8, -, -, -, -, -, -) = Yes

Partial Queries

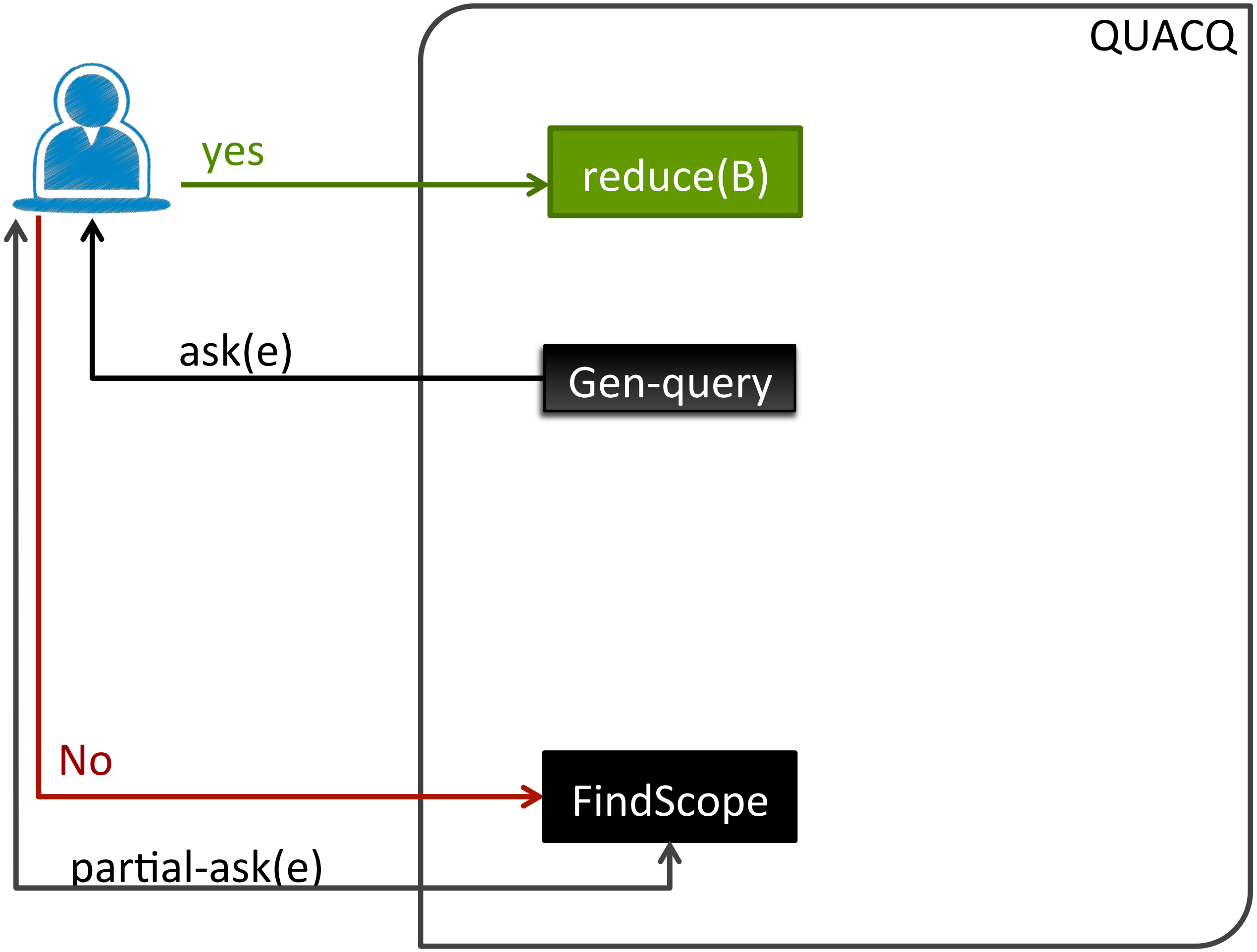


ask(2, 8, 4, -, -, -, -, -) = No

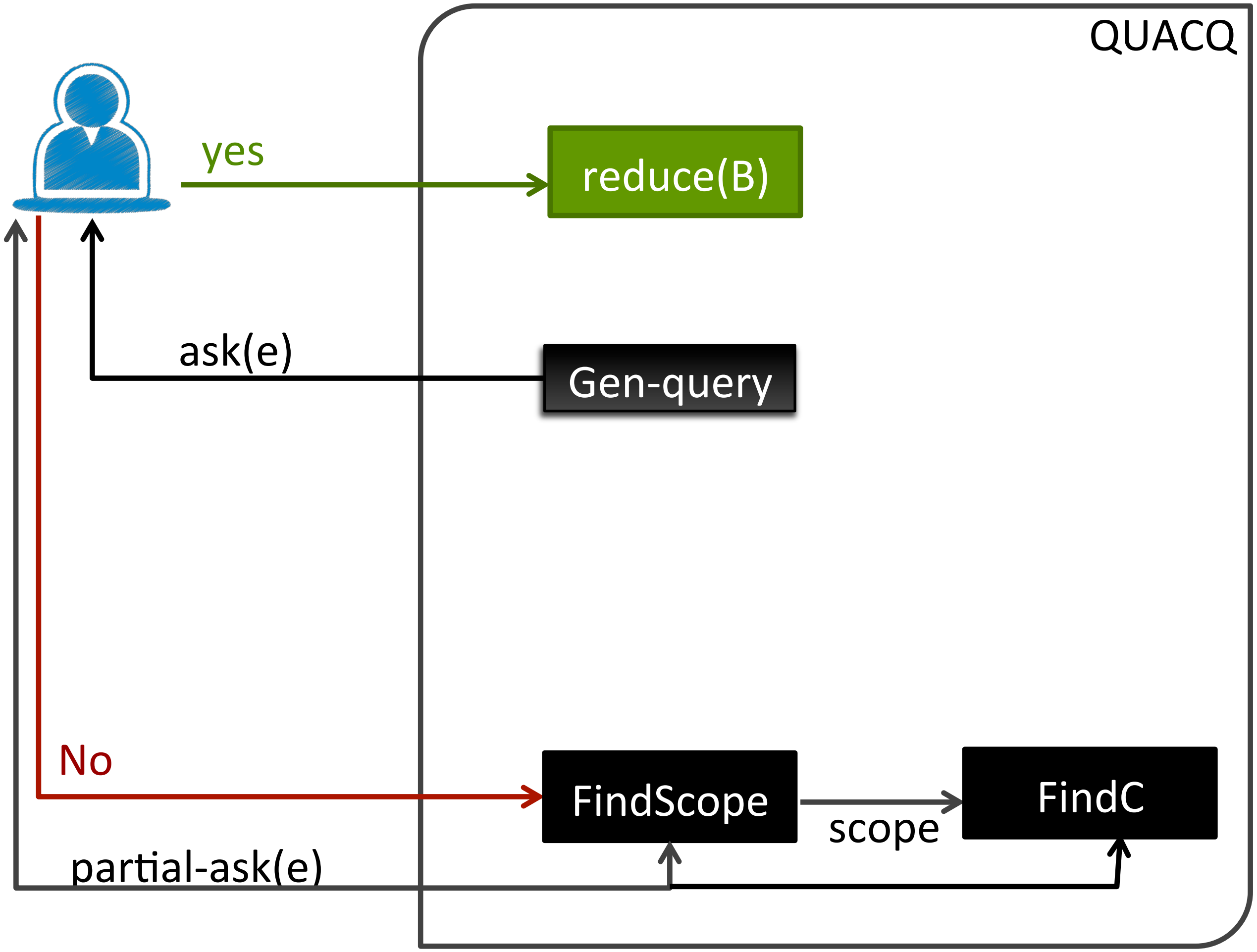
QUACQ: Quick Acquisition



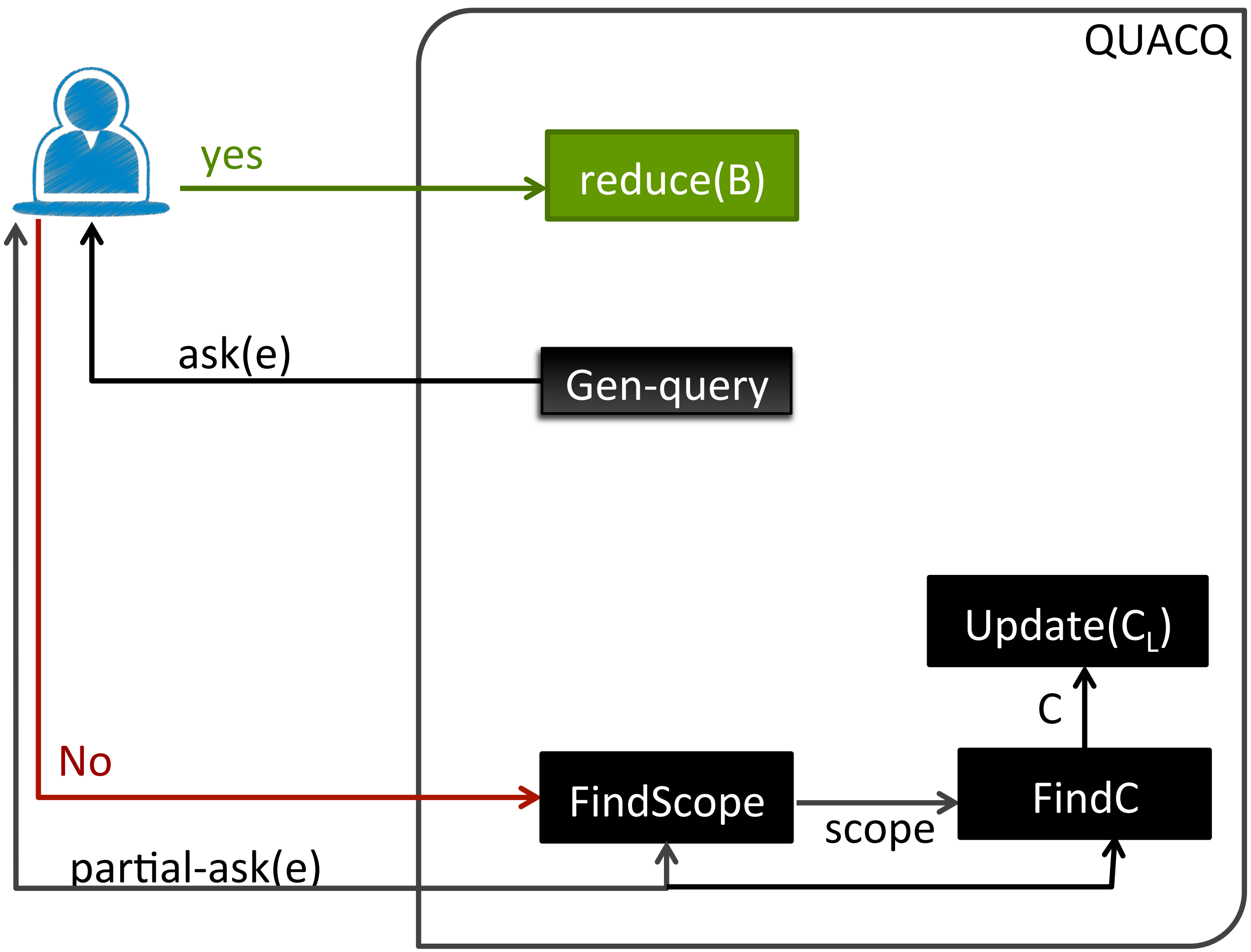
QUACQ: Quick Acquisition



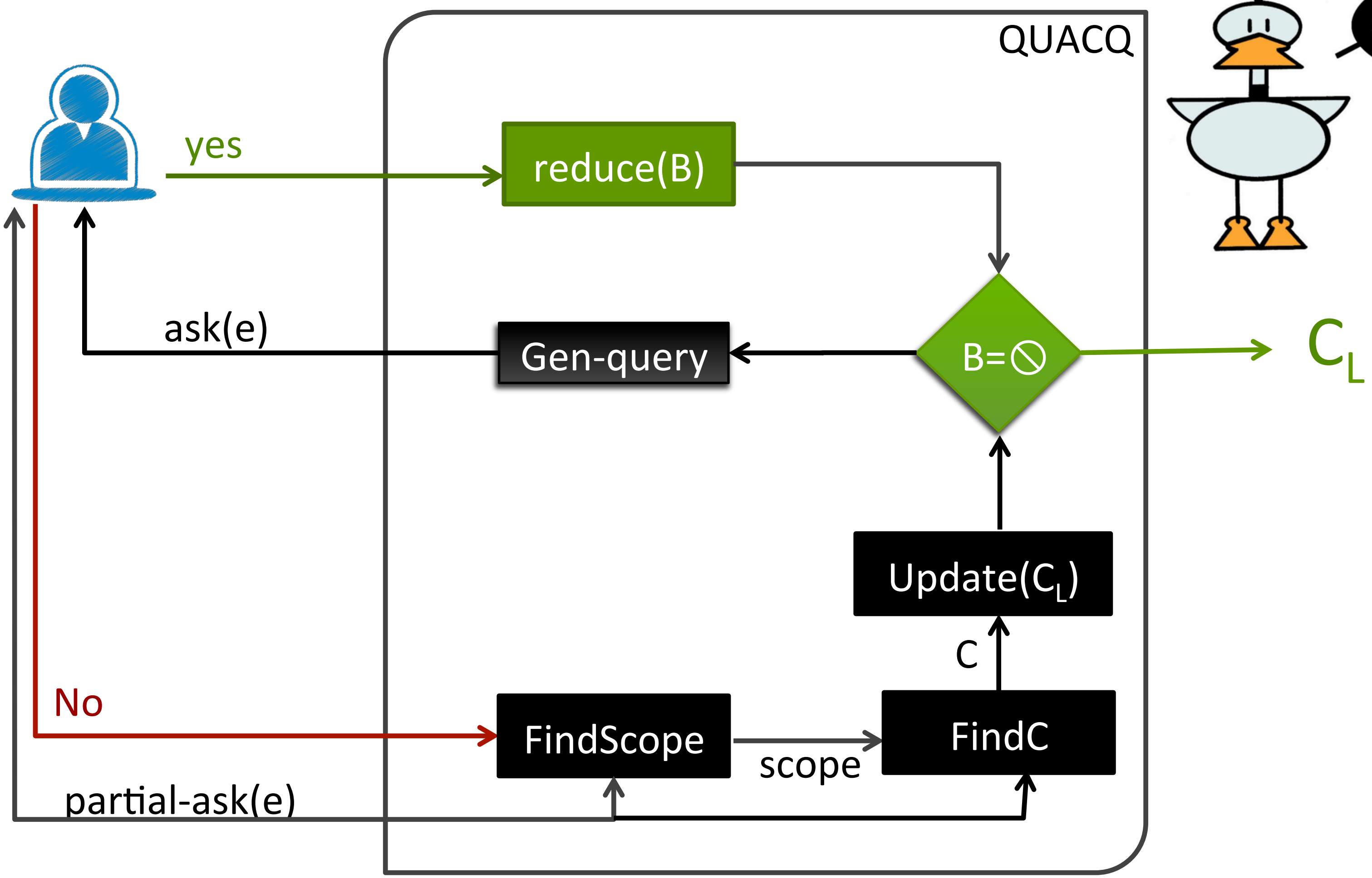
QUACQ: Quick Acquisition

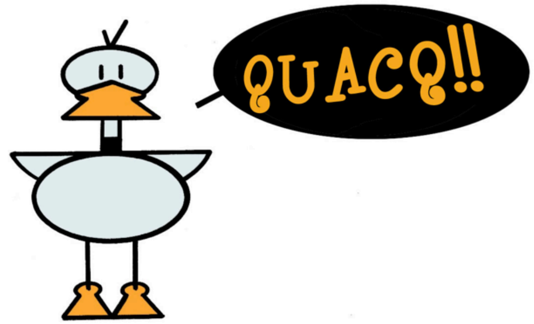


QUACQ: Quick Acquisition



QUACQ: Quick Acquisition





Complexity of QUACQ

- The number of queries required to find the target concept is in:

$$O(|C_T| \cdot (\log |X| + |\Gamma|))$$

- The number of queries required to converge is in:

$$O(|B|)$$

E^-

E^+

Some Results

- Random
  - Under-constrained instance (X,D,C)=(50, 10, 12)
  - Phase transition instance (X,D,C)=(50, 10, 122)
  - |B|= 7350 built on $\Gamma = \{=, \neq, <, \geq, >, \leq\}$

Some Results

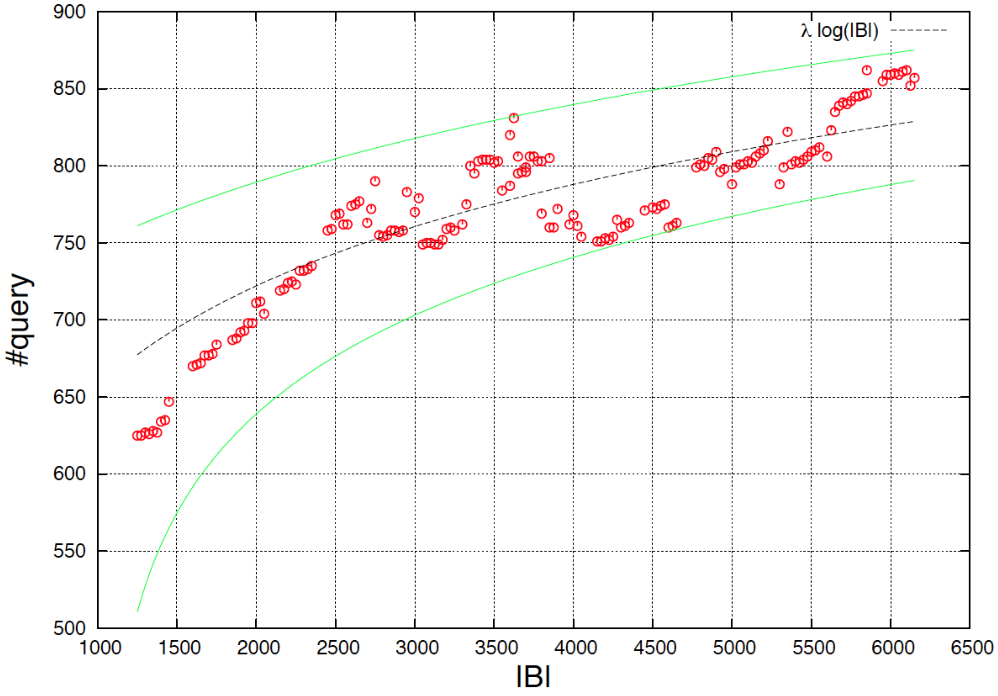
- Random
  - Under-constrained instance (X,D,C)=(50, 10, 12)
  - Phase transition instance (X,D,C)=(50, 10, 122)
  - |B|= 7350 built on $\Gamma = \{=, \neq, <, \geq, >, \leq\}$

	$\lvert C_L \rvert$	$\#q$	$\#q_c$	$\bar{q}$	time
rand_50_10_12	12	196	34	24.04	0.23
rand_50_10_122	86	1074	94	13.90	0.14

Intel Xeon E5462 @ 2.80GHz with 16 Gb of RAM.

Some Results

- Zebra puzzle
  - QUACQ behavior on different bias sizes



Intel Xeon E5462 @ 2.80GHz with 16 Gb of RAM.

Some Results

- Sudoku

A target network on 81 variables with 810 constraints

	$\|C_L\|$	$\#q$	$\#q_c$	$\bar{q}$	time
Sudoku 9×9	810	8645	821	20.58	0.16

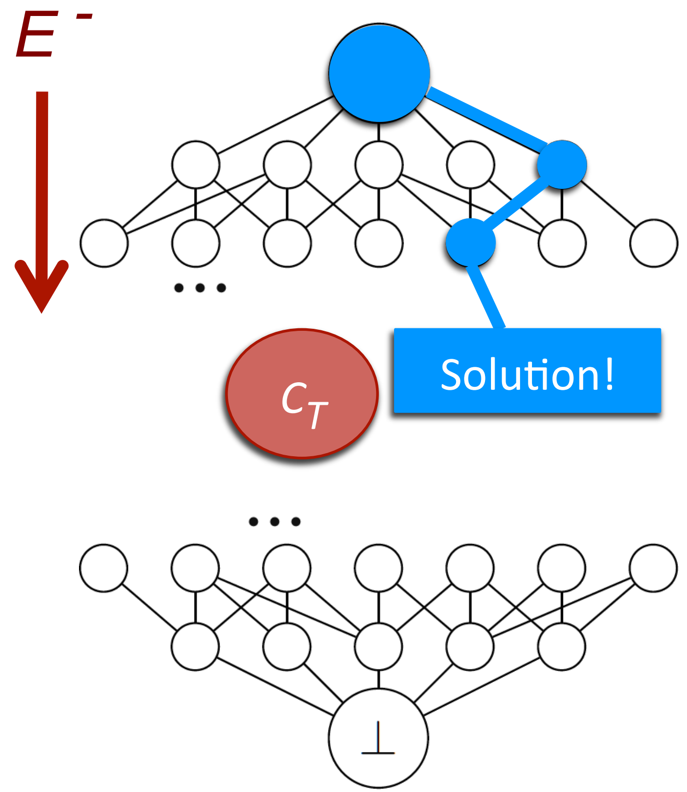
Intel Xeon E5462 @ 2.80GHz with 16 Gb of RAM.

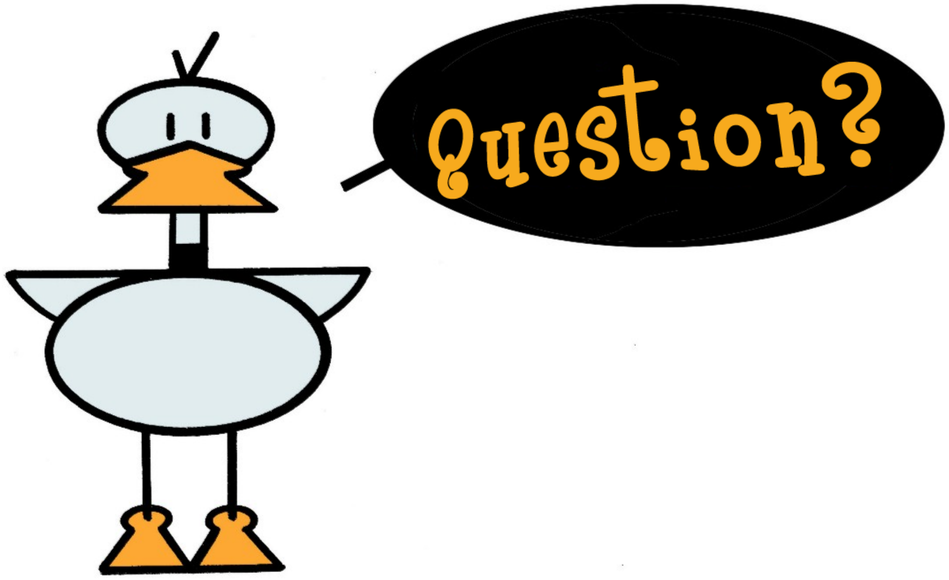
Conclusions

- QUACQ: new constraint acquisition approach based on partial queries
  - Active learning approach
  - Learning a constraint in a log scale of #queries
  - Queries are often much shorter than membership ones
  - Can follow a top-down search to learn a constraint network

Perspectives

- QUACQ as a solver
  - QUACQ does not require positive examples
  - we can use it to ***solve*** an instance
- Ask more than yes/no questions
  - GENACQ for Generalization Acquisition [ECAI14] (next talk!)




Question?