ASP-based Inductive Logic Programming applied to Phrase Chunking: Challenges and Improvements

Peter Schüller

Knowledge Based Systems Group
TU Wien, Wien, Austria

Computer Engineering Department
Marmara University, Istanbul, Turkey

Motivation

- Participation in SemEval 2016 iSTS (Interpretable Semantic Similarity) (Agirre et al., 2016)
- Rule-based (ASP) system for aligning phrases of 2 sentences
- ASP-based ad hoc encoding for Phrase Chunking
  - better than baseline
  - much worse than CRF-based systems
  - 3 very different datasets with 756/750/330 sentences

Motivation

- Participation in SemEval 2016 iSTS (Interpretable Semantic Similarity) (Agirre et al., 2016)
- Rule-based (ASP) system for aligning phrases of 2 sentences
- ASP-based ad hoc encoding for Phrase Chunking
  - better than baseline
  - much worse than CRF-based systems
  - 3 very different datasets with 756/750/330 sentences

$\Rightarrow$ Let's learn rules for chunking from datasets!

ILP Formulation of Chunking

- Instance:
  $pos(Pos, Token)$
- Background Knowledge:
  $postype(Pos) \leftarrow pos(Pos, Token).$
  $token(Token) \leftarrow pos(Pos, Token).$
  $nextpos(Pos, Token{-}1) \leftarrow pos(Pos, Token), token(Token{-}1).$
- Potential Head Atoms in Learned Program (#modeh):
  $split(+token)$
- Potential Body Atoms in Learned Program (#modeb):
  $pos(\$postype, +token)$
  $nextpos(\$postype, +token)$

ILP Examples for Chunking

- Example Sentence

$[\text{Former}_1\ \text{Nazi}_2\ \text{death}_3\ \text{camp}_4\ \text{guard}_5\ \text{Demjanjuk}_6]\ [\text{dead}_7]\ [\text{at}_8\ 91_9]$

- Background Knowledge:

$pos(c\_NNP, 1).\quad pos(c\_NNP, 2).\quad pos(c\_NN, 3).$
$pos(c\_NN, 4).\quad pos(c\_NN, 5).\quad pos(c\_NNP, 6).$
$pos(c\_VBD, 7).\quad pos(c\_IN, 8).\quad pos(c\_CD, 9).$
$goodchunk(1) \leftarrow not\ split(1), not\ split(2), not\ split(3),$
$not\ split(4), not\ split(5), split(6).$
$goodchunk(7) \leftarrow split(6), split(7).$
$goodchunk(8) \leftarrow split(7), notsplit(8).$

- Examples:

$\#example\ goodchunk(1).$
$\#example\ goodchunk(7).$
$\#example\ goodchunk(8).$

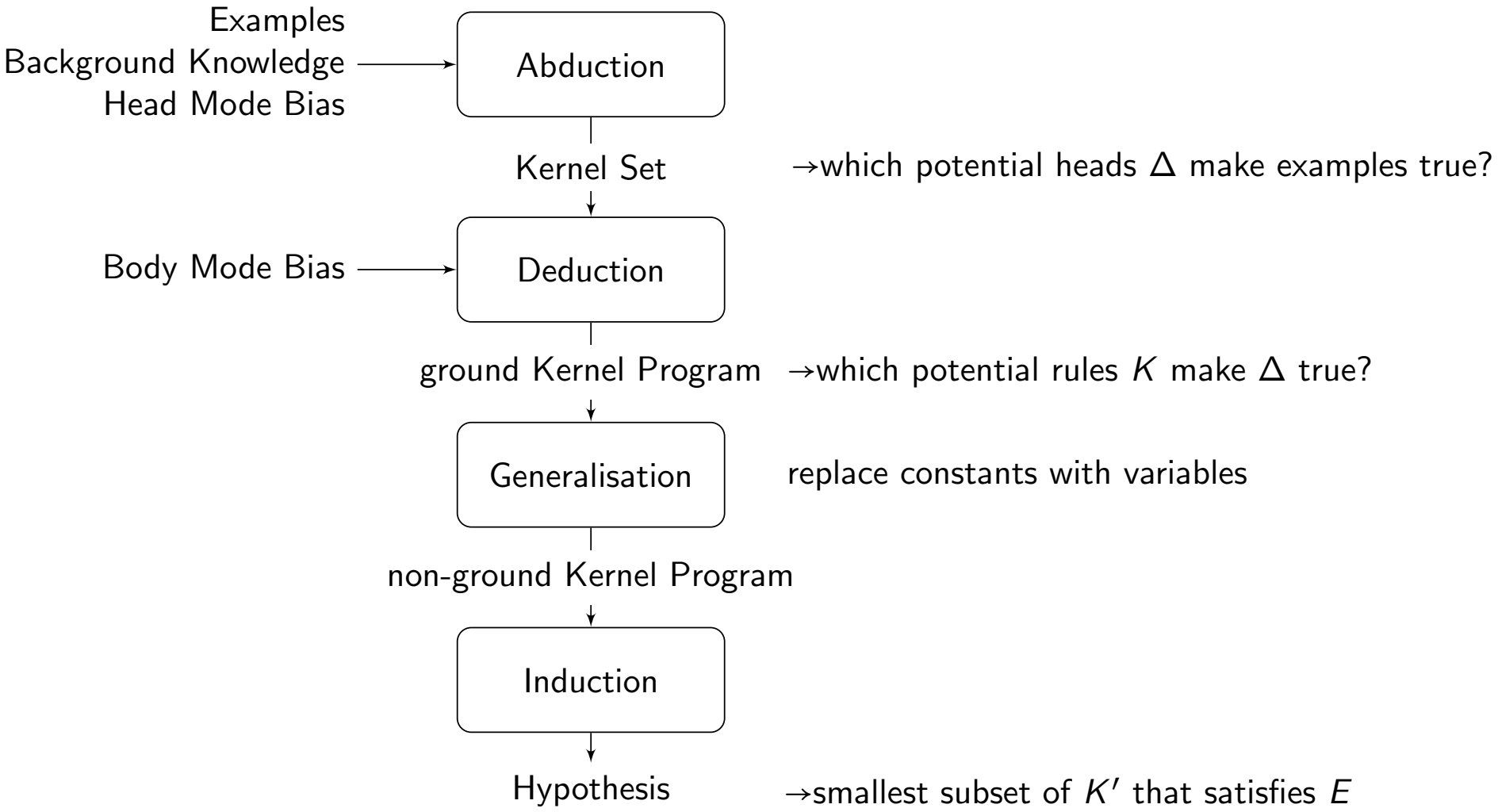
Attempts with Existing Solvers

- XHAIL (Ray, 2009):
  open source, old ASP solver technology
  feasible with 50 examples (<1 day for finding hypothesis)
- ILED (Katzouris et al., 2015):
  open source, incremental solving
  fails for noisy input, several problematic algorithm corner cases
- ILASP2 (Law et al., 2015):
  closed source, most advanced capabilities (learning guesses)
  feasible with 5 examples (<1 day for finding hypothesis)

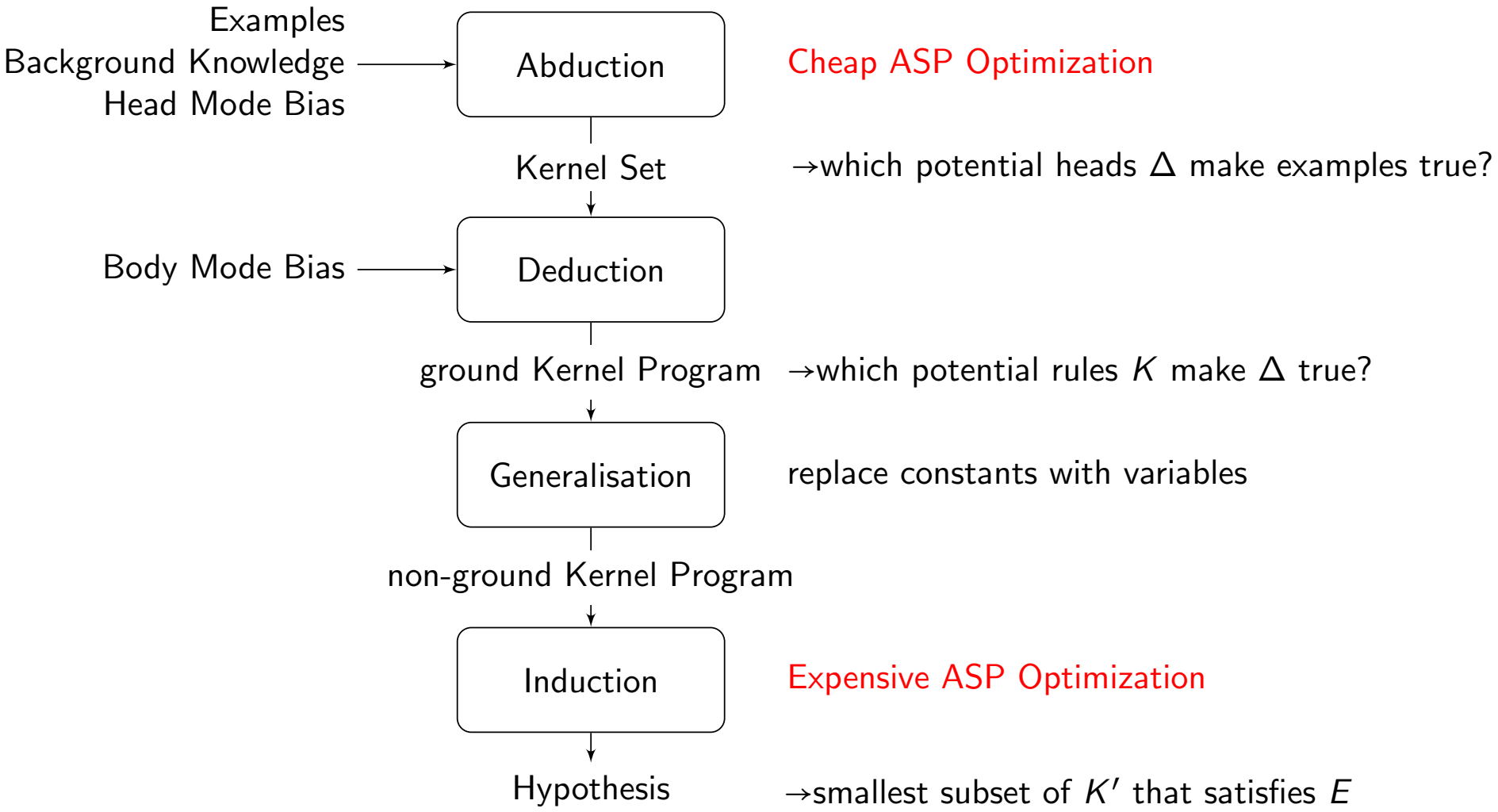
Attempts with Existing Solvers

- XHAIL (Ray, 2009):
  open source, old ASP solver technology
  feasible with 50 examples (<1 day for finding hypothesis)
- ILED (Katzouris et al., 2015):
  open source, incremental solving
  fails for noisy input, several problematic algorithm corner cases
- ILASP2 (Law et al., 2015):
  closed source, most advanced capabilities (learning guesses)
  feasible with 5 examples (<1 day for finding hypothesis)
- Infeasible with ASP-based ILP technology

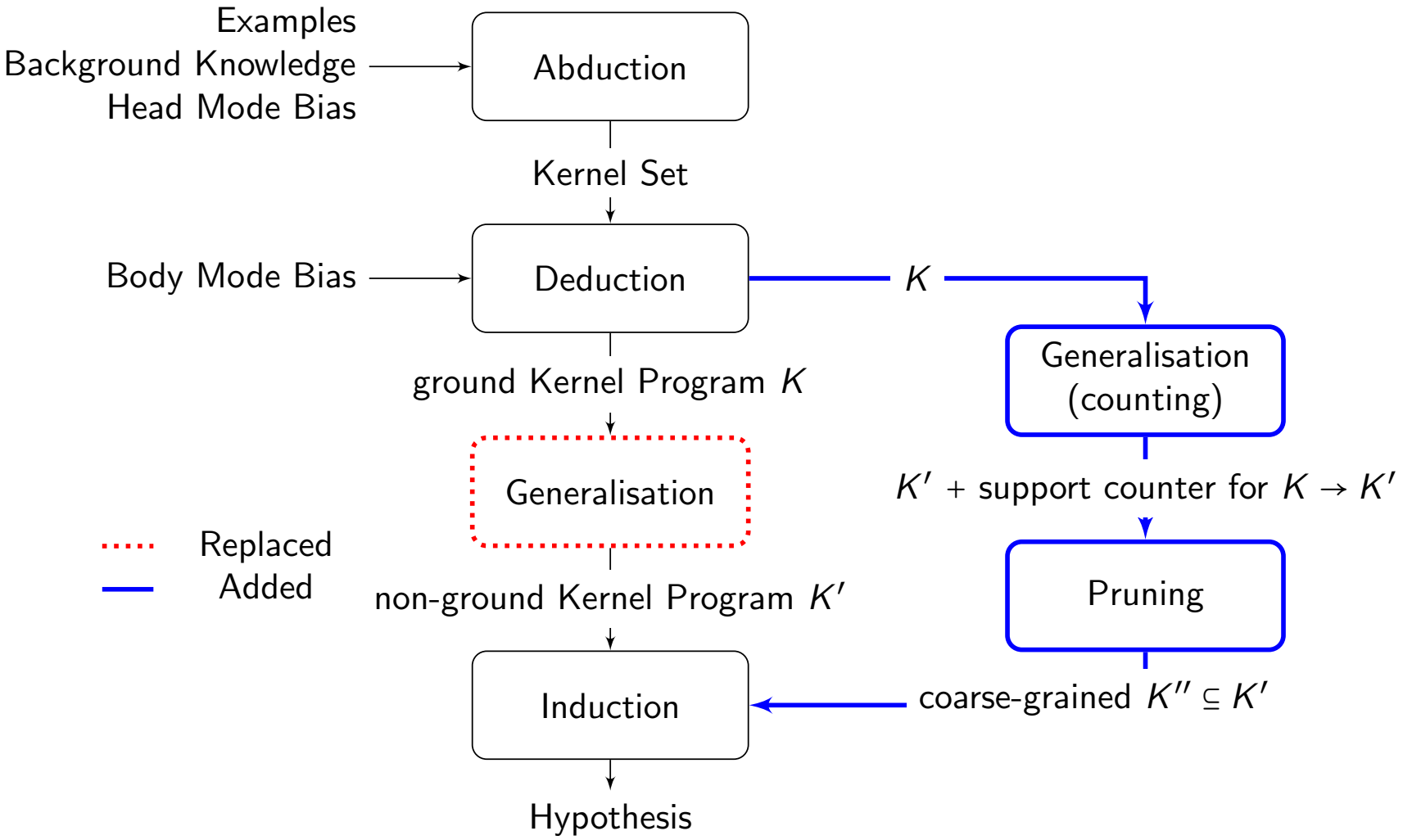
XHAIL Principle (Ray, 2009)



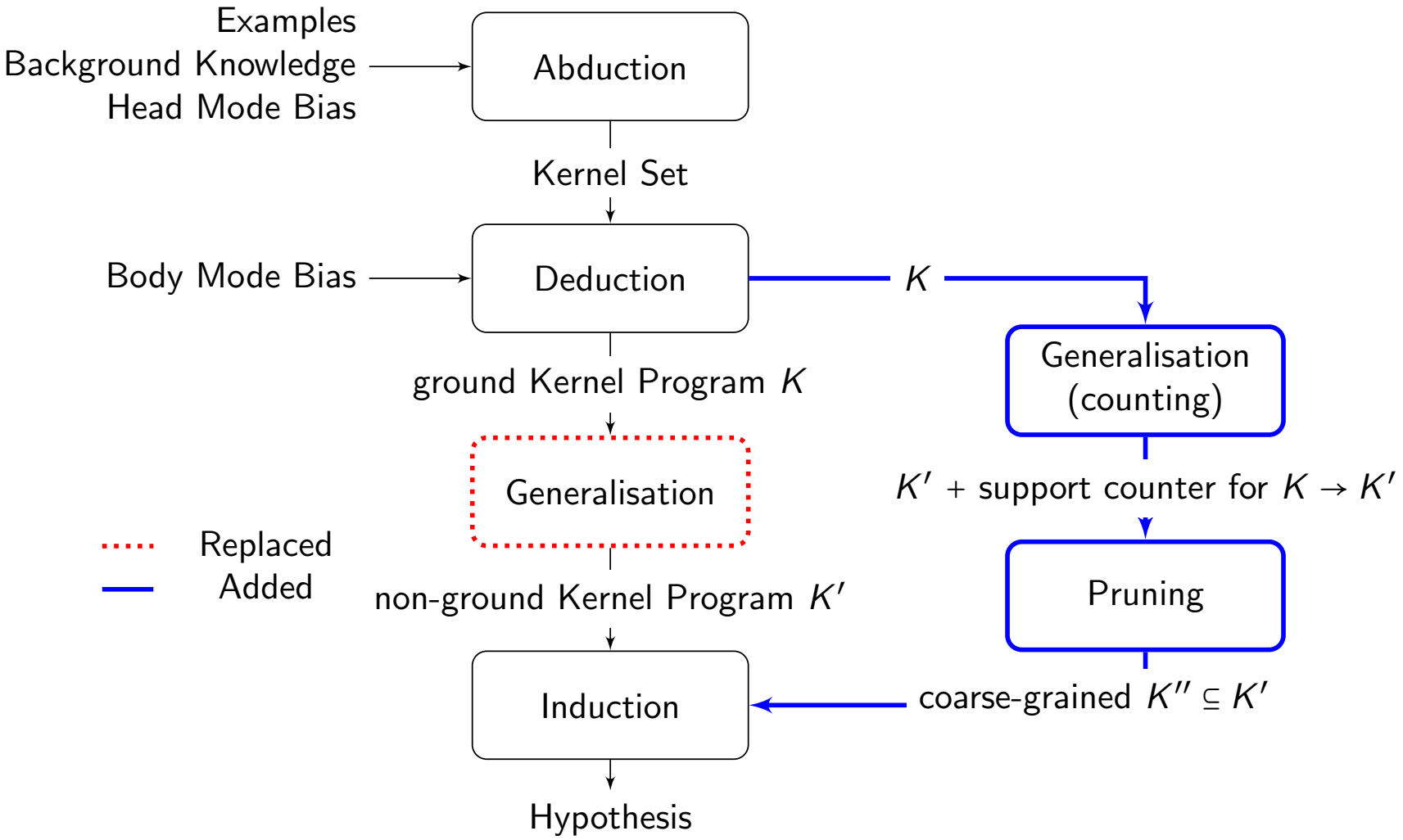
XHAIL Principle (Ray, 2009)



XHAIL Pruning (Kazmi et al., 2017)



XHAIL Pruning (Kazmi et al., 2017)



- K'' = nonground rules that contribute to $\geq P$ examples

Examples	Pruning	Suboptimality	Train F1	Test F1
100	0	172.8	70.7	60.5
	1	10.9	69.3	62.4 $^{\downarrow *}$
	2	0.3	66.5	62.7
	3	0.0	65.1	61.6
500	0	31954.4	39.2	38.9
	1	17855.1	40.7	39.7
	2	6238.5	53.0	49.4
	3	4260.9	59.2	54.7
	4	1598.6	67.1	59.7 $^{\downarrow *}$
	5	1117.2	73.3	62.1 $^{\downarrow *}$
	6	732.6	71.7	62.7 $^{\downarrow *}$
	7	561.4	71.2	63.6
	8	475.0	71.8	64.0
	9	312.7	71.2	63.3
	10	220.3	73.4	63.7

Resource Limits:
30 min, 5 GB

Pruning = adjustment
LoD vs. Optimality

Wrapup

- Pruning and modern ASP algorithms make learning feasible.
- Hypotheses have provided insights about dataset issues.
- F1 score is significantly better than with manual rules.

Wrapup

- Pruning and modern ASP algorithms make learning feasible.
- Hypotheses have provided insights about dataset issues.
- F1 score is significantly better than with manual rules.
- Contribution: parameter for level of detail vs. optimality.
- Limitation: mode bias very simple.

Possible Future Directions

- Incremental learning with ASP + noise.
- Pruning for multi-model hypothesis ILP algorithms.
- Combining Prolog-based and ASP-based ILP techniques.
- Find logic programming fragments that permit scalability and provide (limited) nonmonotonicity.

Possible Future Directions

- Incremental learning with ASP + noise.
- Pruning for multi-model hypothesis ILP algorithms.
- Combining Prolog-based and ASP-based ILP techniques.
- Find logic programming fragments that permit scalability and provide (limited) nonmonotonicity.

Thank you for your attention!

contact Peter Schüller
schueller.p@gmail.com
Knowledge Based Systems Group
Faculty of Informatics
TU Wien, Vienna, Austria

Agirre, Eneko et al. (2016). "SemEval-2016 Task 2: Interpretable Semantic Textual Similarity". In: *International Workshop on Semantic Evaluation (SemEval)*, pp. 512–524. URL: http://www.aclweb.org/anthology/S16-1082.

Katzouris, Nikos et al. (2015). "Incremental learning of event definitions with Inductive Logic Programming". In: *Machine Learning* 100.2-3, pp. 555–585. ISSN: 0885-6125. DOI: 10.1007/s10994-015-5512-1. URL: http://link.springer.com/10.1007/s10994-015-5512-1.

Kazmi, Mishal et al. (2017). "Improving Scalability of Inductive Logic Programming via Pruning and Best-Effort Optimisation". In: *Expert Systems with Applications* 87, pp. 291–303. ISSN: 09574174. DOI: 10.1016/j.eswa.2017.06.013. arXiv: 1706.05171. URL: http://linkinghub.elsevier.com/retrieve/pii/S0957417417304311.

Law, Mark et al. (2015). "Learning Weak Constraints in Answer Set Programming". In: *Theory and Practice of Logic Programming* 15.4-5, pp. 511–525. ISSN: 1471-0684. DOI: 10.1017/S1471068415000198. arXiv: arXiv:1507.06566v1.

Ray, Oliver (2009). "Nonmonotonic abductive inductive learning". In: *Journal of Applied Logic* 7, pp. 329–340. ISSN: 15708683. DOI: 10.1016/j.jal.2008.10.007. URL: http://dx.doi.org/10.1016/j.jal.2008.10.007.