Sample questions for preparing to the oral exam

1) Datalog

Which feature(s) are missing standard relational query languages?

Which approaches are available to define the semantics of Datalog, and how do they work?

What is active domain semantics, and how can it be ensured?

Datalog syntax: how do Datalog rules and programs look like?

How is the least model semantics of Datalog conceptually defined, and why is one choosing the least model?

What is Program Grounding, and what is the role of Herbrand models for Datalog?

What is the difference between Datalog and logic programming, and how can the two be reconciled?

How is fixpoint semantics for Datalog conceptually defined, and which important result of algebra is used for that?

What is the benefit of the fixpoint semantics for Datalog?

What would happen under fixpoint semantics if not active domain semantics would be used?

How is the proof-theoretic semantics for Datalog conceptually defined?

How can proof trees in Datalog be constructed in principle?

Explain how conceptually SLD resolution works. What kind of problems come with it, and how can they be resolved?

Name some Datalog systems

2) Datalog with Deterministic Negation

What problem surfaces when adding negation to Datalog?

What is rule safety, and why is this adopted?

What problems do the fixpoint semantics and least model semantics of Datalog have when it is (naively) extended to negation?

What solution approaches exist to remove these problems?

Explain conceptually what stratified semantics of Datalog is.

How is stratified semantics formally defined (just the idea, not the details)?

What does the Stratification Theorem informally say, and is this important?

How can we test whether a Datalog program is stratified, and how expensive is that (polynomial or intractable)?

What is the idea of three-valued semantics for the evaluation of atoms and rules?

Which two ways for defining the well-founded semantics of Datalog programs did we discuss?

How is the immediate consequence operator T_P conceptually extended in order to define well-founded semantics? What is the technical "recipe" that is used?

What is an unfounded set, and why does a greatest unfounded set exist?

What is the idea to define well-founded semantics using unfounded sets?

Does every program have a well-founded model?

3) ASP Paradigm

What is the idea for the stable model semantics, if one looks at the extended immediate consequence operator T_{P,J} ?

What does one mean with "nondeterminism" in stable semantics?

How is certain semantics informally defined?

Describe the ASP idea (ASP Paradigm)

Which form do disjunctive answer set programs have? Why is disjunction added?

How is the semantics of disjunctive answer set programs defined?

Can disjunction in rule heads, like in in(X) v out(X) :- node(X)., be always easily eliminated, by rewriting the rules?

Given an example of an ASP program that has no answer sets.

Which basic properties of answer sets of a disjunctive answer set program do you know?

How do well-founded semantics and answer set semantics relate to each other?

Suppose given program has a single answer set. Do then answer set semantics and well-founded semantics coincide?

Describe the Guess and Check Methodology

Would a (fixed) stratified ASP program allow for deciding 3-colorability of a graph (which is given as data)?

Would a (fixed) ASP program under well-founded semantics allow for deciding 3-colorability of a graph (which is given as data)?

Name some application areas of ASP. What makes the use of ASP beneficial?

4) ASP Extensions

Which kinds/types of ASP extensions do you know ?

What is strong negation, and is it needed?

What are choice constructs in ASP, conceptually?

Describe the notion of a weak constraint, and how such constraints are informally used for defining program semantics

What is the Guess Check and Optimize Methodology?

What are aggregates, and what is the challenge to provide them in ASP?

How does an aggregate atom look like?

How can answer semantics be extended to programs with aggregates?

5) ASP Computation

Name some reasoning tasks for ASP programs

What complexity does satisfiability for ASP programs (i.e., deciding whether some answer set exist) have for normal logic programs?

What complexity does satisfiability for ASP programs (i.e., deciding whether some answer set exist) have for disjunctive logic programs?

Is under answer set semantics evaluating disjunctive logic programs more difficult than evaluating normal logic programs?

Is the Grounding & Solve approach good or bad from the perspective of worst-case complexity of evaluation?

Name some ASP solvers; what is downscaling?

Name some approached to evaluate ASP programs

What is the predominant evaluation approach for ASP to date?

What is the idea of intelligent grounding?

What is the grounding bottleneck, and how could it be overcome?

Why is model search in ASP more complicated than in SAT solving?

What problem do function symbols bring in ASP, and how can one manage it?

Describe the idea of finitely grounded programs

What is the predominant way of model computation for ASP programs to date?

What is a portfolio-solver for ASP, and which do you know?

What is the Clark Completion of a program, and when can it be used to capture the answer sets of a logic program?

What is informally a loop formula?

What is the key idea for modern ASP solvers?

What is Multi-shot ASP solving, and what can it be useful for?

What benefits does ASP have compared to SAT?

What benefits/drawbacks does ASP have compared to CP

What is an IDE, and which do you know for ASP? What is a big issue for such IDEs still?

6) Description Logics (DL)

What is an ontology from a data management perspective?

Name some application areas of Ontologies

What is SNOWMED?

How do Description Logics relate to First-order Logic?

What does the language of Description Logics usually comprise?

Which concept / role constructors do DLs have beyond Boolean-style constructors?

How is a DL knowledge base usually composed?

Which forms of terminological axioms do you know?

How can the semantics of DLs be defined?

How are interpretations in DL informally defined?

Which concepts are in ALC defined by restriction, and which pitfall exists for the usage in practice?

How is satisfaction of an ALC knowledge base by an interpretation defined?

Name standard reasoning problems in DL, and how they relate to Satisfiability testing.

Name some non-standard reasoning tasks in DLs; can they be reduced to Satisfiability testing easily?

What is SROIQ, and why is it important?

What are lightweight DLs?

What is the idea for the Description Logic EL?

Which normal form we can use for EL?

What is the main idea for canonical model construction in EL?

How can Satisfiability testing be done for EL^\\bot?

What is EL^{++}?

What design rationale is behind DL-Lite?

Where has DL-Lite been successfully been applied?

How do axioms in DL-Lite look like?

What is the idea for the canonical model construction of DL-Lite?

What is data complexity, what is combined complexity?

What is the data complexity of EL, what of DL-Lite?

What is OWL?

Name the OWL-2 profiles and the underlying Dls

7) Existential Rules

When modeling Description Logics in (pure) Datalog, for which axioms issues arise, i.e., it is not easily possible?

How is negation handled in Datalog/ASP vs Description Logics?

What are the main differences between DLs and Datalog with respect to the intended usage?

How can Datalog be extended to cater for modeling of the axioms that can not be modeled in (pure) Datalog? What is the problem with such an extension?

What is the main reasoning service in Datalog^+, and how does it work?

What is the idea of Datalog^{+/-}?

What is the chase procedure and how does it work?

What is a universal model, and how can it be exploited ?

What is the data resp. combined complexity of BCQs for pure Datalog, and how much does that go up in the worst case for the fragments of Datalog^+ that we considered?

What is Linear Datalog, and what data complexity it has?

What is first-order rewritability resp. Datalog rewritability? Can you give examples?

What is guarded Datalog? How does it relate do DLs?

What is Sticky Datalog? Is it more expressive than DL-Lite?

Name some systems that implement reasoning for existential rules.

8) Hybrid Knowledge Bases

What is a hybrid knowledge base?

Which principled ways did we consider for giving a semantics to hybrid knowledge bases?

What is the idea of
- tight semantic integration?
- full semantic integration?
- loose coupling?

give some example(s), and say what advantages/disadvantages the types of combination have

What is informally dl-safety, what weak dl-safety? Why is it useful?

What is the idea for non-monotonic dl-programs?

Which constituents does an dl-atom have?

How does a dl-program differ from an ordinary ASP program?

How are dl-atoms informally evaluated, and how are informally answer sets defined for dl-programs?

What is an FLP-answer set?

What properties do answer sets of dl-programs enjoy?

For what applications is the use of dl-programs handy?

What is "uniform evaluation" of dl-programs, and why is considered? What issues arise with it?

Describe the idea (steps) for the transformation of dl-programs over OWL2 RL ontologies into ASP, as used by DReW.

Would the DReW transformation be applicable to Description Logics like ALC or SROIC?

9) Answer Set Programming - External Evaluation

What are "inside" and "outside" ASP extensions? Can you give examples?

What is embedded ASP?

What is ASP + X, and what examples do you know?

Which issues are especially important for the definition of a external information access from an ASP program?

What is the notion of an external atom in HEX, and what constituents does it have?

How is the semantics of an external predicate &g informally defined? What pragmatic assumptions are made in that, and what is/could be the motivation for that?

How do HEX programs look like, and how are answer sets for a HEX program defined?

Name some properties of answer sets that are lifted from ordinary ASP programs to HEX programs.

Under which assumption about the complexity of evaluating external atoms are HEX programs not harder to evaluate than ordinary ASP programs?

What is computation outsourcing, what information outsourcing? Give some examples

Describe some examples of application areas of HEX programs. Can you imagine one not in the list discussed?

Name some implementation(s) of HEX programs

How is the black-box nature of external atoms countered in HEX?

Give examples of properties of external predicates that can be exploited.

How can value invention in HEX programs be handled?

Name some formalisms related to HEX programs

Describe the differences to clingo (version 5)

10) ASP Stream Reasoning

What is Streaming Data?

What is Stream reasoning, and how does it differ from Temporal
Reasoning?

How is Stream reasoning different from Stream Processing?

What is conceptually a window function? Which types of window functions occur in practice?

What are the elements of the LARS language?

How is satisfaction of a formula in LARS informally defined?

What are LARS programs, and how is informally an answer stream defined?

What features of answer sets are inherited to answer streams?

What is nondeterminism of answer streams, and how can that be exploited?

What is the complexity of deciding whether a LARS program has an answer stream over a given data stream?

How does propositional LARS with sliding time-based windows compare to

- first-order logic
- second-order logic,
- regular languages?

What is complexity-wise the difference between LARS and linear-time
logic (LTL) ?

Name some implementations of LARS, and what features they have

What uses of LARS for Analysis of other Stream reasoning languages
do you know?

What is CQL, what is ETALIS?

What is Complex Event Processing, and where is a principled difference between the semantics of LARS and the semantics of ETALIS?

When can LARS answer streams capture ETALIS minimal models for integer timelines?

What other ASP/Datalog like Streaming reasoning approaches do you know?