Chemistry Studio
An Intelligent Tutoring System

Abhishek Kar, Akshay Mittal, Ankit Kumar & Ashish Gupta

Prof. Amey Karkare
Dr. Sumit Gulwani (MSR Redmond)
Dr. Ashish Tiwari (SRI)

Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

May 16, 2012

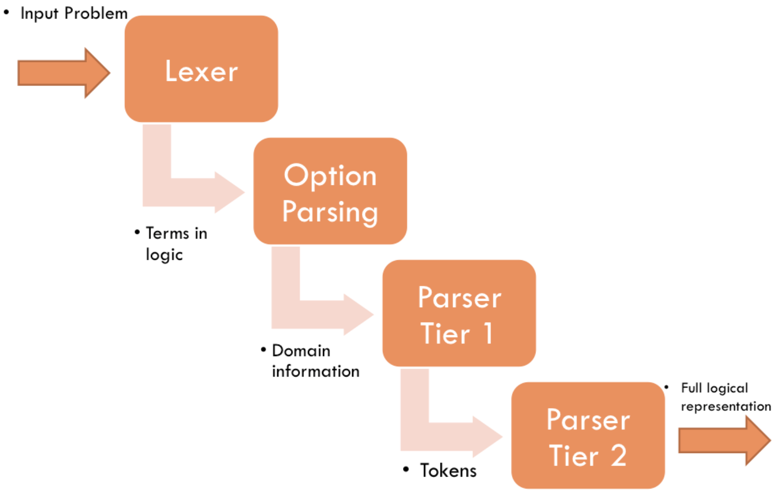
Objective

- Building a system aimed at helping a student in their learning
- Generate solutions and explanations in accordance with the interest and knowledge of the student.
- Hint Generation and Problem Generation
- System emulates the thinking of student
  - In-built knowledge of Basic Facts
  - Breaks complex problem into smaller problems
  - Builds up explanation (using rules)
- Target Users: High school students (Grade 9 to 12)

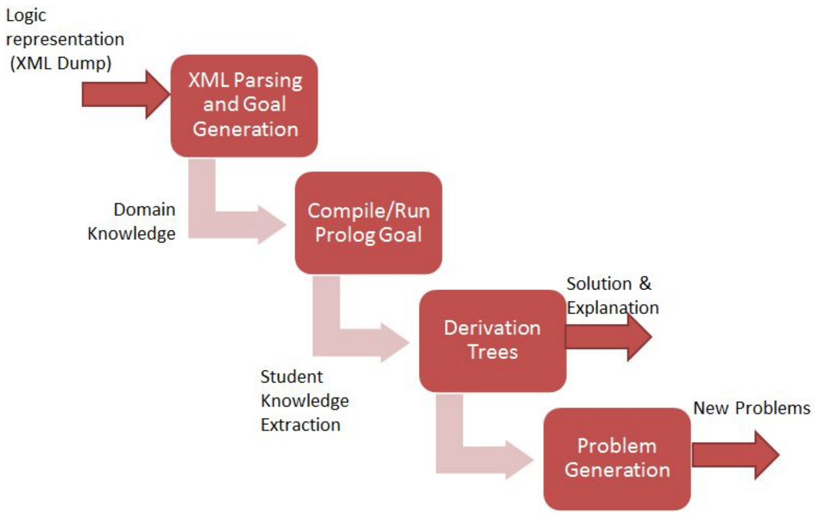
Major Checkpoints

1. Logical formulation of the entities in a Periodic Table.
2. Parsing English language question
3. Generating intermediate logical representation
4. Templates such as Max/Min, Order, Trend, ForAll
5. Derivation Tree
6. Extraction of Student Knowledge
7. Wrong Answer Analysis
8. Generation of New Problems

FLOWCHART: NLP



Flowchart: Problem Solving



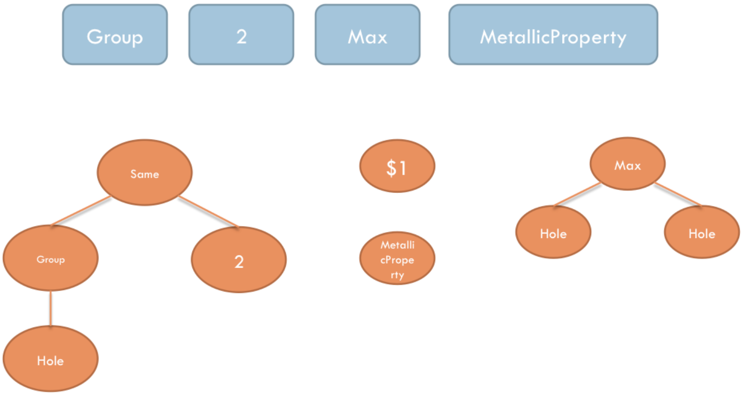
INTERMEDIATE LOGIC COMPONENT

- Identification of major entities in the Periodic Table
- Intermediate representation between the NLP and Problem Solving component.
- Construct a set of facts and rules in the logic programming language Prolog
- Terms: Predicates, Functions, Variables

Unary Predicates	Unary Functions
AlkaliMetals	FirstIonizationEnergy
AlkalineEarthMetals	AtomicRadius
Transition Metals	IonicRadius
Metalloids	AtomicNumber
Non-Metals	GroupNumber
Halogens	MetallicCharacter

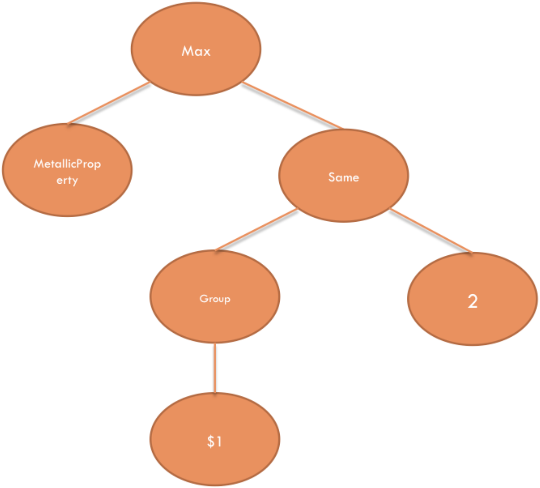
PARSER TIER 1

- Which element in group 2 has the maximum metallic property?
  - i)Be ii)Mg iii)Ca iv)Sr



Parser Tier 2

- *Max*(*MetallicProperty*, *Same*(*Group*($1), 2))



Handling Quantifiers

- Universal Quantifiers: $\forall x\, A(x) \implies B(x)$
- Existential Quantifiers: $\exists x\, A(x) \wedge B(x)$
- Problems
  - Finding the position of implication
  - Finding the antecedent and consequent
- **Example**: Alkali metals show metallic character
  Solution: ForAll($1, Implies(AlkaliMetal($1), Metallic($1)))
- Position of implication $\approx$ Position of verb
- Deciding antecedent and consequent $\rightarrow$ Active vs. Passive voice

Assertion Based Questions

- Assert facts → Pose questions
- Span multiple sentences
- **Example** - An element A forms covalent bond with oxygen. **It** has high electronegativity and belongs to group 13. What is **its** atomic number
- Problem - Anaphora Resolution
- Solution - Use Stanford CoreNLP to find coreference graph
- Stitch solution of the form $A_1(x) \wedge A_2(x) \ldots A_n(x)$, where $A_i(x)$ is the logical formula of the i^{th} sentence and quantify over free variables

Negations

- Non:
  - Which of the following **non**-metals is a gas at STP?
  - Couple non with the predicate immediately next to it
  - *And*(*IsGasAtSTP*($1), *Not*(*Metallic*($1)))
- Not:
  - **Not** all alkali metals form basic oxides.
  - Negation of statement to the right of not.
  - *Not*(*ForAll*($1, *Implies*(*AlkaliMetal*($1), *BasicOxide*($1))))
- No:
  - **No** halogen is metallic in nature.
  - Natural interpretation of **no** as “there does not exist”
  - *Not*(*Exists*($1, *And*(*Halogen*($1), *Metallic*($1))))

Sample Template (ForAll Query)

- Trivial Approach
  - Scan all elements of Periodic Table to find binding
- Our Approach
  - Assert the antecedents with new constants
  - Satisfy the goal with consequents containing new constants

Database

$$A(x), A2(x) \rightarrow P(x)$$

$$A1(x), A2(x) \rightarrow R(x)$$

$$R(x) \rightarrow Q(x)$$

$$P(x), Q(x) \rightarrow B(x)$$

Goal

$$\forall x : A(x), A1(x), A2(x) \rightarrow B(x)$$

Assertions

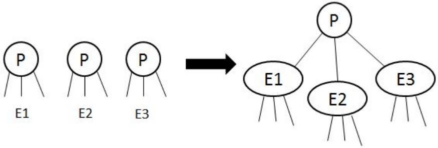
$A(a)$, $A1(a)$, $A2(a)$

New Goal

$B(a)$

Derivation Tree and Student Knowledge

- Can have multiple successful derivations for a single goal
  - Represent different *explanations* to a particular problem
  - Important to let the the user decide which solution to prefer
- Use the existing knowledge-base of the user to simulate which solutions would be more attractive to the user.
  - For example: Solution using Concept *X* and Concept *Y*
  - Ranking is done on the basis of the fraction of the predicates known to user
- Multiple Answers/Explanation per problem
- Combination of different derivation trees to highlight the point of difference



Wrong Answer Analysis

- Relevance
  - Intelligently setting possible options to a given problem
- Closeness of the options to the original answer
  - Minimum number of changes required to make the option correct
  - Depth of the derivation tree with respect to a particular choice
- Observation: Given a predicate with a point failure at the choice variable
  - If all the other arguments in this predicate are bound before its execution, then there is nothing interesting to decide why the option does not satisfy the goal.
  - If there exists at least one argument which will get bound after the execution of the predicate and will be used in later satisfaction of predicates, then that can lead interesting option selection
- Proposed an algorithm to detect all possible points of failure

Problem Generation

- Introduce type information for arguments present in each predicate ("Type Qualifiers" or "Refined Types")

- Every question that is generated by the system can be checked for validity in the following two ways
  - Type-safe checking
  - Presence of at least one solution for the question

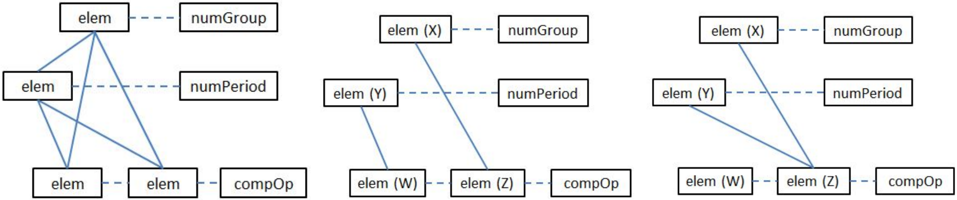
- Level-1 Questions
  - For a predicate $P(arg_1, arg_2, \ldots, arg_n)$, n questions can be generated by binding $(n-1)$ arguments at a time and the binding for the n^{th} argument will serve as a choice variable for the student

- Level–k Questions
  - For a question which maximally utilizes the k-predicates, the predicates should be related to each other via some argument

Problem Generation Algorithm

- Algorithm for generating all possible bindings
  - Each argument of a predicate is a node in the graph
  - Argument nodes of the same predicate and type are connected by a edge
  - Each spanning tree corresponds to a question

- `group(X, 2), period(Y, 3), firstIonisationEnergy(Z, W, ‘+’)`
- $(X = Z), (Y = W) \Rightarrow$
  `Which element in group 2 has ionisation energy greater than an element of period 3 ?`



Future Work

- Adding student database and using ML techniques for feedback
- Paraphrasing of the derivation trees
- Plan to introduce it to the school students
- Problem Generation by example
- Handling Periodic Table Exceptions and Conflicting Rules
- Implementing the proposed combined derivation tree algorithm using database engine such as MySQL