Running the Advanced Prolog Program

Quick Start with SWI-Prolog

1. Install SWI-Prolog

bash

```bash
# Windows: Download installer from https://www.swi-prolog.org/
# macOS:
brew install swi-prolog

# Linux (Ubuntu/Debian):
sudo apt-get install swi-prolog

# Linux (CentOS/RHEL):
sudo yum install pl
```

2. Save and Load the Program

bash

```bash
# Save the code as 'advanced_prolog.pl'
# Start SWI-Prolog
swipl

# In the Prolog prompt, load the file:
?- [advanced_prolog].
```

3. Try These Example Queries

Basic Expert System

prolog

```prolog
% Start forward chaining to learn new facts
?- forward_chain.

% Check what was learned
?- learned(X).

% Prove something about Tweety
?- prove(mortal(tweety)).
```

Constraint Logic Programming

```prolog
% Solve a simple 4x4 Sudoku
?- sudoku([[1,_,_,4],[_,_,_,_],[_,_,_,_],[4,_,_,1]]).

% Solve 4-Queens problem
?- n_queens(4, Solution), label(Solution).
```

Graph Algorithms

```prolog
% Find shortest path from 'a' to 'f'
?- dijkstra(a, f, Path, Distance).

% Find all paths
?- dijkstra(a, e, Path, Distance).
```

Data Structures

```prolog
% Create and search a binary search tree
?- bst_insert(nil, 5-five, T1),
   bst_insert(T1, 3-three, T2),
   bst_insert(T2, 7-seven, T3),
   bst_search(T3, 3, Value).

% Sort a list using quicksort
?- quicksort([3,1,4,1,5,9,2,6,5], Sorted).
```

Natural Language Processing

```prolog
% Parse simple sentences
?- parse_sentence([the, cat, chases, a, mouse], _).
?- parse_sentence([a, dog, sees, the, bird], _).
```

Advanced Analysis

```prolog
% Run comprehensive knowledge base analysis
?- analyze_knowledge_base.

% Benchmark sorting algorithms
?- benchmark_sorting.
```

Troubleshooting

Common Issues:

1. **"ERROR: Syntax error" when loading**
   - Check file encoding (should be UTF-8)
   - Make sure brackets and parentheses match
   - Verify all predicates end with periods
2. **"ERROR: Unknown directive: use_module(library(clpfd))"**
   - This library might not be available in all Prolog systems
   - Try online SWI-Prolog (SWISH) instead
3. **"Warning: clauses of X are not together"**
   - This is just a warning, program will still work
   - To fix: group all clauses of same predicate together

Alternative: Simplified Version

If you encounter library issues, here's a minimal version that works everywhere:

```prolog
% Simple facts
likes(mary, food).
likes(mary, wine).
likes(john, wine).
likes(john, mary).

% Simple rules
happy(X) :- likes(X, wine).
sad(X) :- \+ happy(X).

% Test queries:
% ?- happy(mary).
% ?- sad(john).
% ?- likes(X, wine).
```

Learning Tips

1. **Start small**: Try simple queries first
2. **Use trace**: `?- trace, your_query.` to see execution steps
3. **Check variables**: Use `?- your_query, write(Variable), nl.`
4. **Read error messages**: Prolog gives helpful syntax error info
5. **Use help**: `?- help(predicate_name).` in SWI-Prolog

Next Steps

Once you have it running:

- Modify the facts and rules
- Add your own predicates
- Try combining different features
- Experiment with the constraint solver
- Build your own expert system!

Happy Prologging! 🎯