

Documentação – Nível 3

Sistema Especialista Automotivo em Prolog

1. Introdução

Este módulo implementa um sistema especialista simples para diagnóstico de problemas automotivos.

O sistema utiliza perguntas ao usuário e regras condicionais para determinar possíveis falhas, simulando uma árvore de decisão baseada em respostas de “sim” ou “não”.

Cada sintoma inicia um fluxo diferente de análise, permitindo identificar possíveis causas de falhas comuns em veículos.

2. Base de Conhecimento (Fatos)

```
sintoma(nao_liga).  
sintoma(aquecendo).  
sintoma(fumaca).
```

Esse fatos definem quais tipos de problemas podem ser investigados pelo sistema.

3. Regras de Diagnóstico

Cada regra `problema(X)` representa um fluxo de perguntas e respostas para chegar a um diagnóstico provável.

A lógica é estruturada usando condicionais encadeadas com `-> (então)` e `; (senão)`.

3.1 Diagnóstico: Carro Não Liga

Fluxo:

1. Pergunta se o painel acende ao girar a chave.
2. Se o painel **não acende**, o problema mais provável é **bateria descarregada**.
3. Se o painel **acende**, pergunta se há combustível.
4. Caso tenha combustível, o sistema entende que o problema não pôde ser identificado.
5. Caso **não tenha combustível**, recomenda procurar um mecânico.

Código:

```
problema(nao_liga) :-  
    sintoma(nao_liga),
```

```

write('seu carro acende o painel quando gira a chave? '),
read(R1),
( R1 == nao ->
    writeln('Provavelmente voce tem um problema de bateria.')
;
R1 == sim ->
    writeln('verifique se seu carro tem combustivel.'),
    read(R2),
    ( R2 == sim ->
        writeln('descobrimos seu problema.')
    ;
    R2 == nao ->
        writeln('recomendo levar a um mecanico.')
    )
).

```

3.2 Diagnóstico: Carro Aquecendo

Fluxo:

1. Verifica se o reservatório de água está baixando.
2. Se estiver baixando, o sistema pergunta se há vazamento embaixo do carro.
3. Com vazamento → possíveis causas:
 - Mangueiras com folga ou furos
 - Furo no radiador
4. Sem vazamento → verifica se o óleo está claro ou marrom claro (água misturada com óleo).
5. Caso afirmativo → possível junta do cabeçote queimada.
6. Caso negativo → água possivelmente entrando na câmara de combustão.
7. Se a água **não** está baixando → recomenda levar a um mecânico.

(A documentação se mantém fiel ao fluxo original do seu código.)

3.3 Diagnóstico: Fumaça no Escapamento

Fluxo:

1. Verifica se o óleo está abaixando.
2. Se estiver abaixando, solicita a verificação das velas para detectar óleo.
3. Se houver óleo nas velas → provável desgaste de anéis e pistões.
4. Se não houver → provável desgaste nos retentores das válvulas.
5. Se o óleo **não está abaixando**, o sistema recomenda apenas procurar um mecânico.

Trecho do código correspondente:

```

problema(fumaca) :-
    sintoma(fumaca),
    writeln('primeiro veja se o oleo do carro esta abaixando'),
    read(R1),
    ( R1 == sim ->
        writeln('retire as velas do carro e veja se ha oleo nas velas'),
        read(R2),
        ( R2 == sim ->
            writeln('oleo entrando na camara de combustao por desgaste nos
aneis ou pistoes.')
        ;
        R2 == nao ->
            writeln('possivel problema nos retentores das valvulas.')
        )
    ;
    R1 == nao ->
        writeln('nao foi possivel diagnosticar, leve a um mecanico.')
    ).
```

4. Funcionamento Geral

O sistema especialista desenvolvido demonstra:

- Construção de árvores de decisão usando regras Prolog.
 - Uso de condicionais encadeadas (-> e ;).
 - Intereração com o usuário via `read/1`.
 - Aplicação de lógica baseada em sintomas.
 - Diagnóstico simples baseado em regras e perguntas.
-