

# Horn-Clauses DSL

Guilherme Azevedo Horn (247127)

Kuan Aprigio Estevão (260205)

André Ribeiro do Valle Pereira(244185)

# Contexto

- Permitir a definição de fatos e regras lógicas de forma declarativa, com uma sintaxe próxima da linguagem natural da lógica proposicional.
- Permite declarar fatos e regras, usando and, or e implicação =>
- Permite aplicar funções aos fatos

# Knowledge Base (KB)

- Banco de conhecimento com todas as regras e fatos
- Formato da regras : (premissas . conclusão)
- Formato dos fatos : (Fato1 Fato2 Fato3 ...)

# Mudar a KB

- É possível mudar a kb, adicionando novas regras ou novos fatos. Na verdade
- Adicionando fatos : Uma nova kb é criada, copiando as regras e adicionando os novos fatos
- Adicionando Regras : Uma nova kb é criada, copiando os fatos e adicionando as novas regras

# Inferência

- Como mencionado anteriormente, a inferência consiste em : dado um objetivo(goal) checar se é possível derivá-lo com a kb definida
- Aqui, a inferência é uma função iterativa que aplica regras e coloca as novas conclusões na kb. Com essas novas conclusões, uma nova iteração é feita até (1) uma conclusão ser o goal ou (2) não houver mais novas regras para concluir

# Sintaxe - Exemplos

- Definindo dois Fatos : (define-facts (A B))
- Definindo a regra A or B => C : (define-rule (=> (or A B) C))
- Verifica se um novo D fato pode ser inferido : (infer D)

# Exemplo

**Código :**

```
(define-facts (A B))
```

```
(define-rule (=> (and A B) C))
```

```
(define-rule (=> C D))
```

```
(define-rule (=> (or D E) F))
```

```
(infer F)
```

**Passos :**

0) fatos iniciais = (A B)

1) A and B -> C, fatos = (A B C)

2) C -> D, fatos (A B C D)

3) D or E -> F, fatos (A B C D F)

4) F está nos fatos