

Grupo 1.

1.

```
estado_inicial(e
    (
        player(a),
        jog(a, [(1,1), (2,3), (3,4), (4,5), (4,6)]),
        jog(b, [(2,2), (1,3), (2,6), (3,5), (5,6)]),
        field(1,2)
    )).

sucessor(e(jog(X), Tab), menos(Linha, N), e(jog(X1), Tab1)):-
    member(linha(Linha,_), Tab), % gera a linha a escolher
    retirar_linha(Tab, Linha, N, Tab1),
    outro(X,X1).

sucessor(e(player(J), jog(J, Lista, P), jog(K, L, Q), field(X1, Y1)),
    colocar(X, Y)
    e(player(K), jog(J, ListaN, 0), jog(K, L, Q), field(X2, Y2))
):-
    member((X, Y), Lista),
    (
        ( X = X1, X2 is Y, Y2 is Y1 );
        ( X = Y1, X2 is X1, Y2 is Y );
        ( Y = X1, X2 is X, Y2 is Y1 );
        ( Y = Y1, X2 is X1, Y2 is X );
    ),
    retirar(Lista, (X, Y), ListaN).

sucessor(e(player(J), jog(J, Lista, P), jog(K, L, Q), F),
    passar
    e(player(K), jog(J, Lista, 1), jog(K, L, Q), F)
).

sucessor(e(player(J), jog(K, L, Q), jog(J, Lista, P), field(X1, Y1)),
    colocar(X, Y)
    e(player(K), jog(K, L, Q), jog(J, ListaN, 0), field(X2, Y2))
):-
    member((X, Y), Lista),
    (
        ( X = X1, X2 is Y, Y2 is Y1 );
        ( X = Y1, X2 is X1, Y2 is Y );
        ( Y = X1, X2 is X, Y2 is Y1 );
        ( Y = Y1, X2 is X1, Y2 is X );
    ),
    retirar(Lista, (X, Y), ListaN).

sucessor(e(player(K), jog(J, Lista, Q), jog(K, L, P), F),
    passar
    e(player(J), jog(J, Lista, Q), jog(K, L, 1), F)
).
```

2.

```
estado_terminal(E):- utilidade(E,_).

utilidade(e(player(J),_,jog(K,[],_),_), -1):- J != K.
utilidade(e(player(J),jog(J,[],_),_,_), +1).

utilidade(e(_,jog(_,_,1),jog(_,_,1),_), 0).
```

3.

Grupo 2.

1.

```
carga(10).
carga(9).
...
carga(1).

max_carga(10).

dist(2,1,9).
dist(1,2,9).

dist(2,3,5).
dist(3,2,5).

dist(1,3,14).
dist(3,1,14).

estar(1).
estar(2).
estar(3).

acao(
    andar,
    [carga(N), estar(S1)],
    [carga(N1), estar(S2)],
    [carga(N), estar(S1)]
):-
    dist(S1,S2,L),
    N1 is N - L.

acao(
    carregar,
    [carga(N)],
    [carga(N1)],
    [carga(N)]
```

```
):-  
    max_carga(N1).
```

2.

```
inicio([estar(1), carga(10)]).  
  
objetivo([estar(3)]).
```

3.

Ir da Sala 1 para a Sala 2, carregar e por fim ir da Sala 2 para a Sala 3.

4.

Começa por ir tentar ir da sala 1 para a 3, falha porque não encontra condições. Vai para a sala 2. Da sala 2 tenta ir para a sala 3, falha. Carrega. Vai para a sala 3.