

# Domínio Agrícola

**Grupo 06:**

**Danilo César, Diógenes Júnior, Gabriel Mariano e Rodrigo Edmar**



# Agricola

O Agricola é um jogo de tabuleiro baseado em turnos onde você é um fazendeiro, e deve administrar uma fazenda.

Dentro do jogo você pode fazer diversas ações como recolher recursos (barro, madeira ou pedra), construir cercas, coletar animais, alimentar os animais. Além disso, você pode melhorar as construções para assim poder aumentar sua família e aumentar a quantidade de trabalhadores disponíveis.



**O domínio**



# Principais tipos e predicados

O domínio conta com diversos tipos para representar os trabalhadores, o round, os recursos, os animais, etc.

```
(:types  
  actiontag goods stage round worker improvement roundclass phaseclass roundparts resource room num - obj  
  buildtag animaltag vegtag gentag - actiontag  
  animal vegetable - goods  
)
```

# RoundClass

Define o tipo de round, oque limita as ações que podem ser tormadas.

- Tnormal: round normal, sem colheita.
- Tharvest: round com colheita.

# ActionTag

Define o tipo de ação que pode, ou será tomada.

# Resource

O domínio possui 4 tipos de recursos:

- Wood (madeira)
- Clay (Barro)
- Reed (cana)
- Stone (pedra)

# Animals

- Sheep (ovelha)
- Boar (porco)
- Cattle (Vaca)

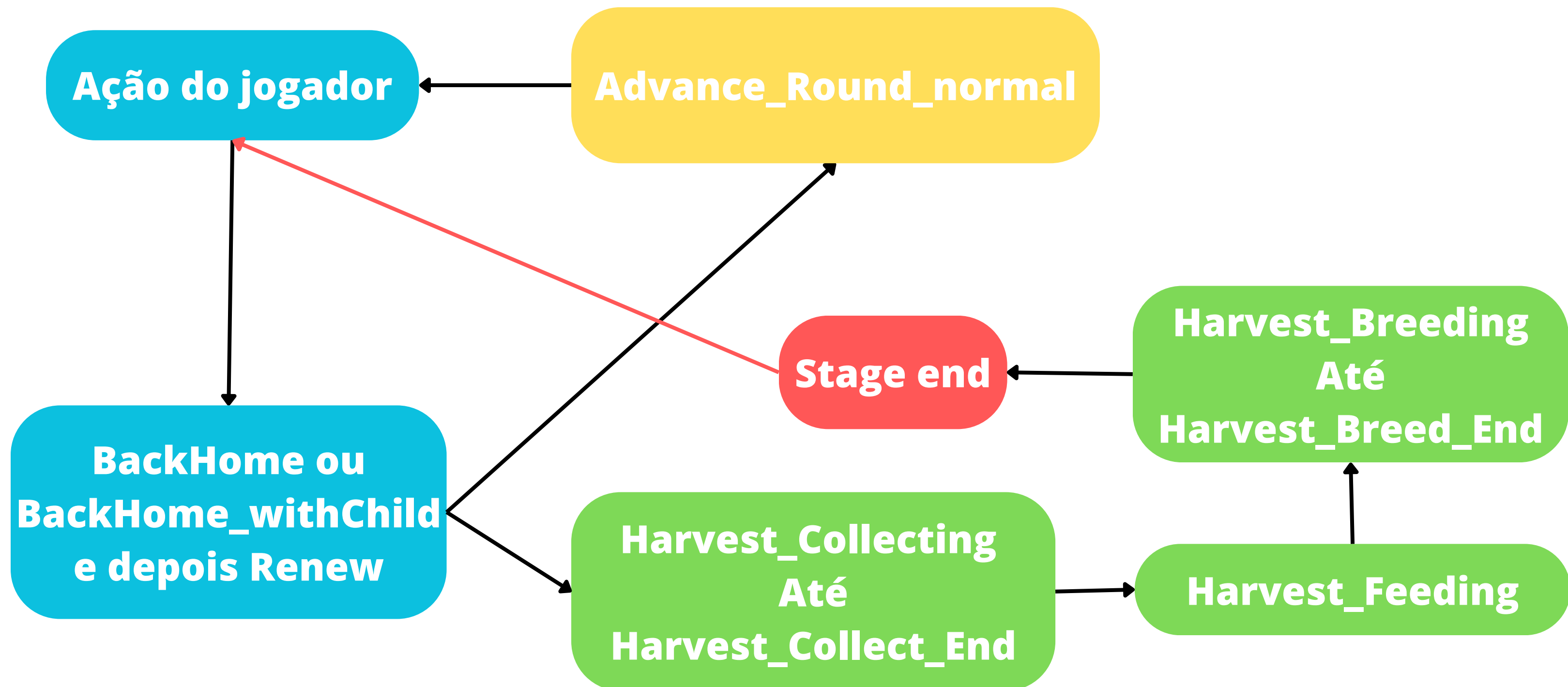
**Ações**

# Ações

O domínio possui dois tipos de ações, as ações executadas pelo **agrícola** durante o turno, e as ações executadas pelo **Player**. Ao todo são **22** ações, sendo **11** para os players e **11** para o Agrícola.

- action **ag**\_\_harvest\_collect\_end (Agrícola)
- action take\_food (player)

# Fluxo do Domínio





## ag\_finish\_round\_backhome

**:parameters** (?r - **round** ?maxw - **worker**)

**:precondition**

(and

(current\_round ?r)

(**current\_worker noworker**)

(max\_worker ?maxw)

(**not** (**newborn**))

)

**:effect**

(and

(**not** (**current\_worker noworker**))

(**current\_worker ?maxw**)

(not (current\_round ?r))

(**hold\_round ?r backhome**)

(increase (total-cost) 1)

)

*Reseta os  
trabalhadores  
disponíveis, se tiver  
filhos ativa o  
backhome\_withchild*

## ag\_finish\_round\_renew

**:parameters** (?r - **round** ?maxw - **worker**)

**:precondition**

(and

(**hold\_round** ?r **backhome**)

)

**:effect**

(and

(**not** (**hold\_round** ?r **backhome**))

(**hold\_round** ?r **roundend**)

**available\_action+**

(can\_harvest grain)

(can\_harvest carrot)

(increase (total-cost) 1)

)

*Permite que o player  
execute ações  
denovo.*

## collect\_resource

**:parameters** (?w1 ?w2 ?wmax - **worker** ?r - **round** ?act - **buildtag**  
?res - **resource**)

**:precondition**

(and

(**available\_action** ?act)

(open\_action ?act)

(current\_worker ?w1)

(**SUPPLY\_RESOURCE** ?act ?res)

**+++** )

**: effect**

(and

(**not** (**available\_action** ?act))

(**not** (**current\_worker** ?w1))

(**current\_worker** ?w2)

(**stored\_resource** ?res)

(increase (total-cost) (**group\_worker\_cost** ?wmax))

)

# Problemas e Execuções

# Problemas

- Foram apresentados um conjunto de 20 problemas diferentes para cada tipo do domínio Agrícola: SAT ou OPT.
- Os problemas dos tipos SAT e OPT diferem em algumas características e parâmetros. Por exemplo, alguns problemas do tipo SAT possuem menos *estágios/fases*, rounds, *trabalhadores e quartos*.
- Para cada execução dos problemas de tipo SAT, foram apresentados os valores inferiores e superiores de custo.
- Para os problemas de tipo OPT, foram apresentados os valores de custo ótimos para cada um.

# Problemas

## SATISFICING TRACK

- single CPU core
- **8Gb** memory limit
- **30min** time limit
- Multiple plans can be returned, the one with the lowest cost is counted.
- The score of a planner on a solved task is the **ratio  $C^*/C$**  where  $C$  is the cost of the cheapest discovered plan and  $C^*$  is the cost of a reference plan. The score on an unsolved task is 0. The score of a planner is the sum of its scores for all tasks.
- If an invalid plan is returned, all tasks in the domain are counted as unsolved.
- If that happens in more than one domain, the entry is disqualified.

## OPTIMAL TRACK

- single CPU core
- **8Gb** memory limit
- **30min** time limit
- Plans must be optimal
- The score of a planner is the **number of solved tasks**
- If a suboptimal or invalid plan is returned, all tasks in the domain are counted as unsolved.
- If that happens in more than one domain, the entry is disqualified.

# Tabela de Execução (IPC18 - SAT)

| ID   | Problem  | Lower Bound | Upper Bound |
|------|----------|-------------|-------------|
| 4767 | p01.pddl | 3275        | 3275        |
| 4774 | p02.pddl | 4004        | 4004        |
| 4771 | p03.pddl | 4142        | 4733        |
| 4781 | p04.pddl | 4218        | 5099        |
| 4776 | p05.pddl | 4102        | 5466        |
| 4764 | p06.pddl | 3931        | 5831        |
| 4763 | p07.pddl | 4414        | 6197        |
| 4779 | p08.pddl | 4197        | 6564        |
| 4778 | p09.pddl | 4232        | 6929        |
| 4765 | p10.pddl | 3966        | 7295        |
| 4762 | p11.pddl | 911         | 911         |
| 4780 | p12.pddl | 1008        | 1008        |
| 4777 | p13.pddl | 897         | 897         |
| 4773 | p14.pddl | 914         | 914         |
| 4772 | p15.pddl | 965         | 1099        |
| 4768 | p16.pddl | 1015        | 1286        |
| 4766 | p17.pddl | 1469        | 2161        |
| 4775 | p18.pddl | 1764        | 2303        |
| 4770 | p19.pddl | 1776        | 2871        |
| 4769 | p20.pddl | 1710        | 2883        |

# Tabela de Execução (IPC18 - OPT)

| ID   | Problem  | Lower Bound | Upper Bound |
|------|----------|-------------|-------------|
| 4547 | p01.pddl | 786         | 786         |
| 4554 | p02.pddl | 878         | 878         |
| 4551 | p03.pddl | 788         | 788         |
| 4561 | p04.pddl | 946         | 946         |
| 4556 | p05.pddl | 979         | 979         |
| 4544 | p06.pddl | 1038        | 1038        |
| 4543 | p07.pddl | 1048        | 1048        |
| 4559 | p08.pddl | 897         | 897         |
| 4558 | p09.pddl | 923         | 923         |
| 4545 | p10.pddl | 1114        | 1114        |
| 4542 | p11.pddl | 1116        | 1116        |
| 4560 | p12.pddl | 1057        | 1057        |
| 4557 | p13.pddl | 1182        | 1182        |
| 4553 | p14.pddl | 974         | 974         |
| 4552 | p15.pddl | 1061        | 1272        |
| 4548 | p16.pddl | 1309        | 1309        |
| 4546 | p17.pddl | 1100        | 1100        |
| 4555 | p18.pddl | 1005        | 1200        |
| 4550 | p19.pddl | 1376        | 1376        |
| 4549 | p20.pddl | 1169        | 1389        |



# Execução em Ambiente Local

- Foram realizadas as execuções de alguns problemas com o domínio **Agrícola** em ambiente local (**chococino**)
- A partir da execução destes com os *planners* disponíveis no ambiente, foram obtidos diversos planos para os problemas testados
- Os custos dos planos gerados variaram de acordo com o tipo do problema (OPT ou SAT) e as *flags* utilizadas na execução dos planejadores

# Primeira Execução

- *Planner*: fast-downward.py
  - *Alias*: lama-first
- Domínio: *Agricola Domain* (OPT)
- Problema: p01.pddl (OPT)
- Comando executado: `/home/software/planners/downward/fast-downward.py --alias lama-first ./domains/opt/agricola/domain.pddl ./domains/opt/agricola/p01.pddl`
- Tempo de Execução: 4.4s
- Custo: 1115

```

(collect_resource worker2 worker1 worker2 round1 act_clay clay)
(collect_resource worker1 noworker worker2 round1 act_reed reed)
(ag_finish_round_backhome round1 worker2)
(ag_finish_round_renew round1 noworker)
(ag_advance_round_normal round1 round2 act_sheep)
(collect_resource worker2 worker1 worker2 round2 act_clay clay)
(collect_resource worker1 noworker worker2 round2 act_reed reed)
(ag_finish_round_backhome round2 worker2)
(ag_finish_round_renew round2 noworker)
(ag_advance_round_normal round2 round3 act_sow)
(collect_resource worker2 worker1 worker2 round3 act_clay clay)
(take_food worker1 noworker worker2 round3 num2 num3)
(ag_finish_round_backhome round3 worker2)
(ag_finish_round_renew round3 noworker)
(ag_advance_round_normal round3 round4 act_fences)
(take_food worker2 worker1 worker2 round4 num3 num4)
(build_fences boar worker1 noworker worker2 round4)
(ag_finish_round_backhome round4 worker2)
(ag_finish_round_renew round4 noworker)
(ag_harvest_collect_end round4 stage1)
(ag_harvest_feed round4 stage1 worker2 num4 num4 num0)
(ag_harvest_breed_end round4 stage1)
(ag_finish_stage stage1 stage2 round4 round5 act_boar)
(take_food worker2 worker1 worker2 round5 num0 num1)
(collect_resource worker1 noworker worker2 round5 act_stone stone)
(ag_finish_round_backhome round5 worker2)
(ag_finish_round_renew round5 noworker)
(ag_advance_round_normal round5 round6 act_improve)
(improve_home worker2 worker1 worker2 round6 fireplace)
(take_food worker1 noworker worker2 round6 num1 num2)
(ag_finish_round_backhome round6 worker2)
(ag_finish_round_renew round6 noworker)
(ag_advance_round_normal round6 round7 act_cattle)
(build_fences boar worker2 worker1 worker2 round7)
(collect_cook_animal boar act_boar worker1 noworker worker2 round7 num2 num4)
(ag_finish_round_backhome round7 worker2)
(ag_finish_round_renew round7 noworker)
(ag_harvest_collect_end round7 stage2)
(ag_harvest_feed round7 stage2 worker2 num4 num4 num0)
(ag_harvest_breed_end round7 stage2)
(ag_finish_stage stage2 stage3 round7 round8 act_carrot)
(collect_cook_animal boar act_cattle worker2 worker1 worker2 round8 num0 num2)
(collect_cook_animal boar act_sheep worker1 noworker worker2 round8 num2 num4)
(ag_finish_round_backhome round8 worker2)
(ag_finish_round_renew round8 noworker)
(ag_advance_round_normal round8 round9 void)
(build_fences boar worker2 worker1 worker2 round9)
(collect_resource worker1 noworker worker2 round9 act_clay clay)
(ag_finish_round_backhome round9 worker2)
(ag_finish_round_renew round9 noworker)
(ag_harvest_collect_end round9 stage3)
(ag_harvest_feed round9 stage3 worker2 num4 num4 num0)
(ag_harvest_breed_end round9 stage3)
; cost = 1115 (general cost)

```

```

ag__harvest_feed round7 stage2 worker2 num4 num4 num0 (1)
ag__harvest_breed_end round7 stage2 (1)
ag__finish_stage stage2 stage3 round7 round8 act_carrot (1)
collect_cook_animal boar act_cattle worker2 worker1 worker2 round8 num0 num2 (60)
collect_cook_animal boar act_sheep worker1 noworker worker2 round8 num2 num4 (60)
ag__finish_round_backhome round8 worker2 (1)
ag__finish_round_renew round8 noworker (1)
ag__advance_round_normal round8 round9 void (1)
build_fences boar worker2 worker1 worker2 round9 (60)
collect_resource worker1 noworker worker2 round9 act_clay clay (60)
ag__finish_round_backhome round9 worker2 (1)
ag__finish_round_renew round9 noworker (1)
ag__harvest_collect_end round9 stage3 (1)
ag__harvest_feed round9 stage3 worker2 num4 num4 num0 (1)
ag__harvest_breed_end round9 stage3 (1)
[t=0.481585s, 33852 KB] Plan length: 53 step(s).
[t=0.481585s, 33852 KB] Plan cost: 1115
[t=0.481585s, 33852 KB] Expanded 112 state(s).
[t=0.481585s, 33852 KB] Reopened 0 state(s).
[t=0.481585s, 33852 KB] Evaluated 123 state(s).
[t=0.481585s, 33852 KB] Evaluations: 236
[t=0.481585s, 33852 KB] Generated 606 state(s).
[t=0.481585s, 33852 KB] Dead ends: 10 state(s).
[t=0.481585s, 33852 KB] Number of registered states: 123
[t=0.481585s, 33852 KB] Int hash set load factor: 123/128 = 0.960938
[t=0.481585s, 33852 KB] Int hash set resizes: 7
[t=0.481585s, 33852 KB] Search time: 0.0707655s
[t=0.481585s, 33852 KB] Total time: 0.481585s
Solution found.
Peak memory: 33852 KB
Remove intermediate file output.sas
search exit code: 0

INFO      Planner time: 4.40s

```

# Demais Execuções

## Caso de Teste 02

- *Planner*: fast-downward.py
- *Alias*: lama-first
- Domínio: *Agricola Domain* (SAT)
- Problema: p01.pddl (SAT)
- Comando Executado: `/home/software/planners/downward/fast-downward.py --alias lama-first ./domains/sat/agricola/domain.pddl ./domains/sat/agricola/p01.pddl`
- Tempo de Execução: 55.59s
- Custo: 3275

## Caso de Teste 03 (próxima página)

- *Planner*: fast-downward.py
- *Alias*: lama
- Domínio: *Agricola Domain* (OPT)
- Problema: p01.pddl (OPT)
- Comando Executado: `/home/software/planners/downward/fast-downward.py --alias lama ./domains/opt/agricola/domain.pddl ./domains/opt/agricola/p01.pddl`
- Tempo de Execução: > 1h31min
- Menor Custo Encontrado: 786

(collect\_resource worker2 worker1 worker2 round1 act\_reed reed)  
(collect\_resource worker1 noworker worker2 round1 act\_wood wood)  
(ag\_finish\_round\_backhome round1 worker2)  
(ag\_finish\_round\_renew round1 noworker)  
(ag\_advance\_round\_normal round1 round2 act\_sheep)  
(build\_room worker2 worker1 worker2 worker3 round2 room3)  
(family\_growth worker1 noworker worker2 worker3 round2 clay room3)  
(ag\_finish\_round\_backhome\_withchild round2 worker2 worker3)  
(ag\_finish\_round\_renew round2 noworker)  
(ag\_advance\_round\_normal round2 round3 act\_sow)  
(take\_grain worker3 worker2 worker3 round3 carrot)  
(collect\_resource worker2 worker1 worker3 round3 act\_clay clay)  
(collect\_resource worker1 noworker worker3 round3 act\_reed reed)  
(ag\_finish\_round\_backhome round3 worker3)  
(ag\_finish\_round\_renew round3 noworker)  
(ag\_advance\_round\_normal round3 round4 act\_fences)  
(take\_food worker3 worker2 worker3 round4 num2 num3)  
(plow\_field worker2 worker1 worker3 round4)  
(sow worker1 noworker worker3 round4 carrot)  
(ag\_finish\_round\_backhome round4 worker3)  
(ag\_finish\_round\_renew round4 noworker)  
(ag\_harvest\_collecting\_veg round4 stage1 carrot num3 num5 num6)  
(ag\_harvest\_collect\_end round4 stage1)  
(ag\_harvest\_feed round4 stage1 worker3 num6 num6 num0)  
(ag\_harvest\_breed\_end round4 stage1)  
(ag\_finish\_stage stage1 stage2 round4 round5 act\_boar)  
(collect\_resource worker3 worker2 worker3 round5 act\_wood wood)  
(build\_room worker2 worker1 worker3 worker4 round5 room4)  
(family\_growth worker1 noworker worker3 worker4 round5 clay room4)  
(ag\_finish\_round\_backhome\_withchild round5 worker3 worker4)  
(ag\_finish\_round\_renew round5 noworker)  
(ag\_advance\_round\_normal round5 round6 act\_improve)  
(take\_food worker4 worker3 worker4 round6 num0 num1)

(collect\_resource worker3 worker2 worker4 round6 act\_stone stone)  
(improve\_home worker2 worker1 worker4 round6 fireplace)  
(collect\_cook\_animal boar act\_sheep worker1 noworker worker4 round6 num1 num3)  
(ag\_finish\_round\_backhome round6 worker4)  
(ag\_finish\_round\_renew round6 noworker)  
(ag\_advance\_round\_normal round6 round7 act\_cattle)  
(collect\_cook\_animal boar act\_sheep worker4 worker3 worker4 round7 num3 num5)  
(take\_grain worker3 worker2 worker4 round7 carrot)  
(take\_food worker2 worker1 worker4 round7 num5 num6)  
(collect\_cook\_animal boar act\_boar worker1 noworker worker4 round7 num6 num8)  
(ag\_finish\_round\_backhome round7 worker4)  
(ag\_finish\_round\_renew round7 noworker)  
(ag\_harvest\_collect\_end round7 stage2)  
(ag\_harvest\_feed round7 stage2 worker4 num8 num8 num0)  
(ag\_harvest\_breed\_end round7 stage2)  
(ag\_finish\_stage stage2 stage3 round7 round8 act\_carrot)  
(take\_grain worker4 worker3 worker4 round8 grain)  
(collect\_cook\_animal boar act\_cattle worker3 worker2 worker4 round8 num0 num2)  
(collect\_cook\_animal boar act\_sheep worker2 worker1 worker4 round8 num2 num4)  
(take\_food worker1 noworker worker4 round8 num4 num5)  
(ag\_finish\_round\_backhome round8 worker4)  
(ag\_finish\_round\_renew round8 noworker)  
(ag\_advance\_round\_normal round8 round9 void)  
(take\_food worker4 worker3 worker4 round9 num5 num6)  
(build\_fences boar worker3 worker2 worker4 round9)  
(collect\_cook\_animal boar act\_sheep worker2 worker1 worker4 round9 num6 num8)  
(collect\_resource worker1 noworker worker4 round9 act\_clay clay)  
(ag\_finish\_round\_backhome round9 worker4)  
(ag\_finish\_round\_renew round9 noworker)  
(ag\_harvest\_collect\_end round9 stage3)  
(ag\_harvest\_feed round9 stage3 worker4 num8 num8 num0)  
(ag\_harvest\_breed\_end round9 stage3)  
; cost = 786 (general cost)

# Referências

**INTERNATIONAL PLANNING COMPETITION.** “INTERNATIONAL PLANNING COMPETITION 2018 CLASSICAL TRACKS”. Disponível em: <<https://ipc2018-classical.bitbucket.io/>>. Acesso em 20 set. 2023.

**INTERNATIONAL PLANNING COMPETITION.** “INTERNATIONAL PLANNING COMPETITION 2018 CLASSICAL TRACKS DOMAINS”. Disponível em: <<https://ipc2018-classical.bitbucket.io/domains.html/>>. Acesso em 20 set. 2023.

**COVIL DOS JOGOS.** “Agricola Edição Revisada - Regras e Gameplay” por Covil dos Jogos. Disponível em: <<https://www.youtube.com/watch?v=VB1gcZWn7UI&t=2487s>>. Acesso em 21 set. 2023.

**LUDOPEDIA.** “Agricola (2007)”. Disponível em: <<https://ludopedia.com.br/jogo/agricola>>. Acesso em 21 set. 2023.



**OBRIGADO**

