# A Planning Graph Heuristic for Forward-Chaining Adversarial Planning

Pascal Bercher and Robert Mattmüller

Institute for Computer Science
University of Freiburg

ECAI 2008, Patras, Greece

Wednesday, July 23, 2008

# Introduction

## Motivation

- Given: Adversarial planning problem (extensive two-player game)
- Desired: Strong plan (winning strategy)

## Technically

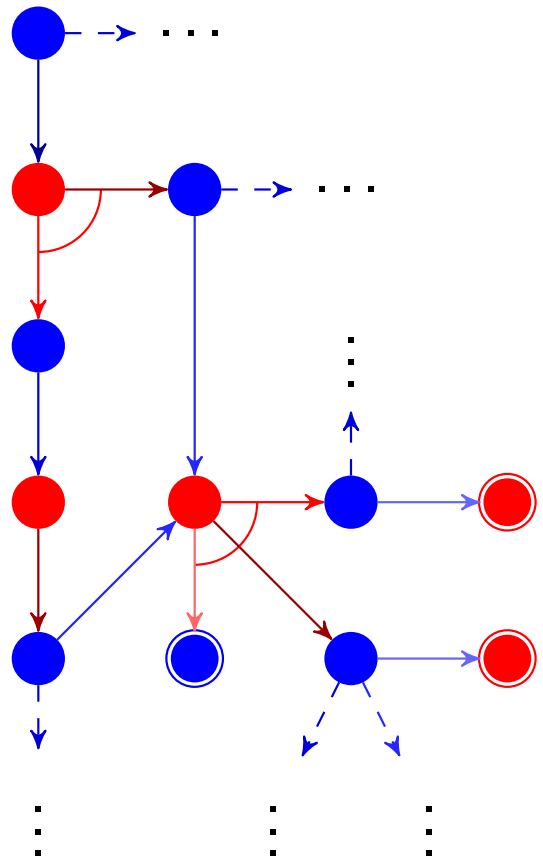
- Two players taking turns
- STRIPS-style state and action encoding
- Full observability
- Reduces to evaluation of AND/OR graph over physical states

# Example

## Problem

- Logistics-like problem
- Pilot and co-pilot have different capabilities (loading, unloading, flying, re-fuelling, no-ops)
- Co-pilot wants to sabotage transport task

## AND/OR Graph and Solution



# Search

- Alternatives:
  - Symbolic regression search (cf. MBP)
  - Heuristically guided explicit-state progression search
- Here: Variant of AO* algorithm
  - Search over AND/OR *graph*
  - Elimination of duplicate nodes
  - Approximative updates of cost estimates
- How to initialize cost estimates at leaf nodes?
  Variant of FF heuristic.

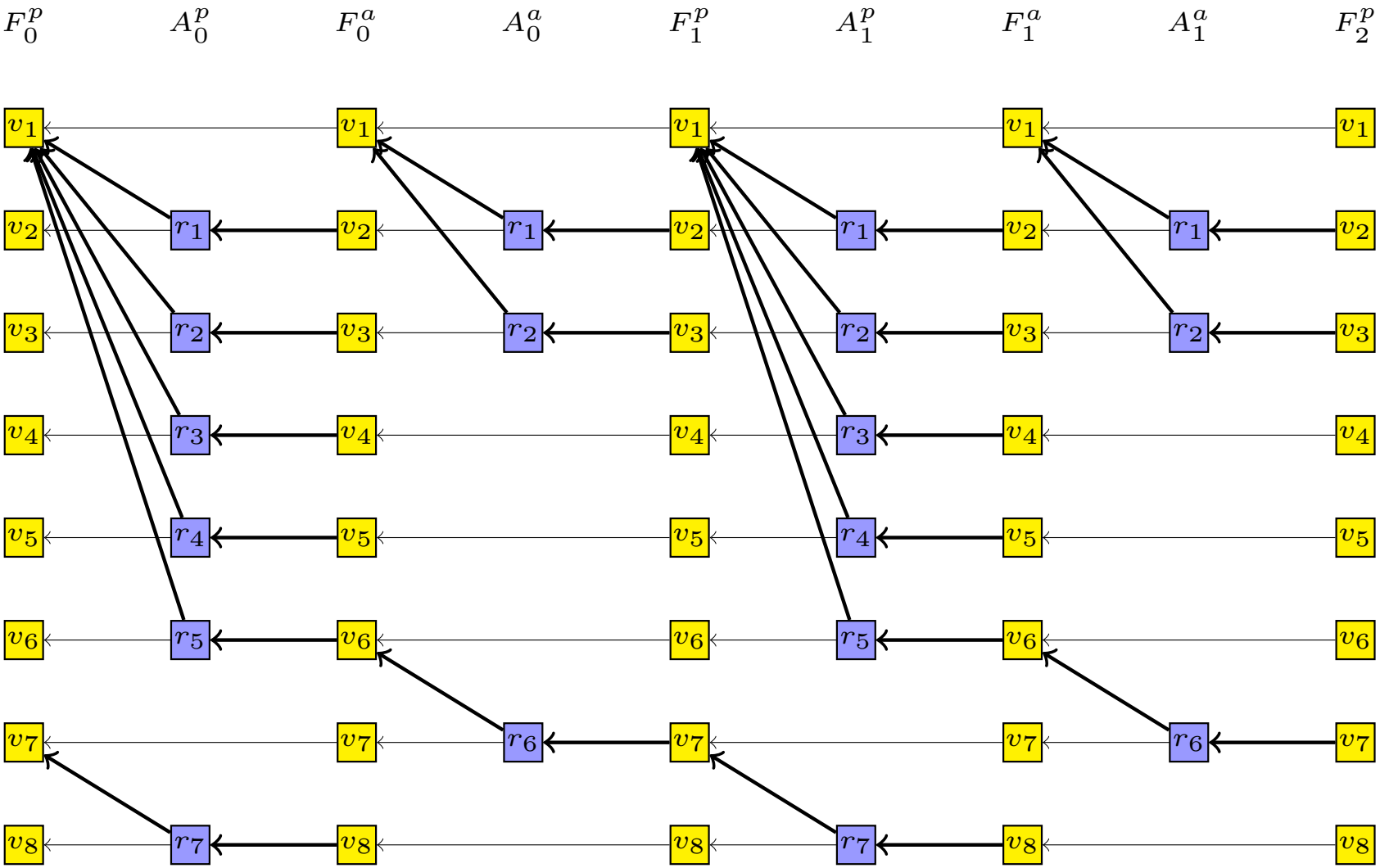
# Heuristic: Example

- Variables: $v_1, \ldots, v_8$
- Rules in relaxed problem:

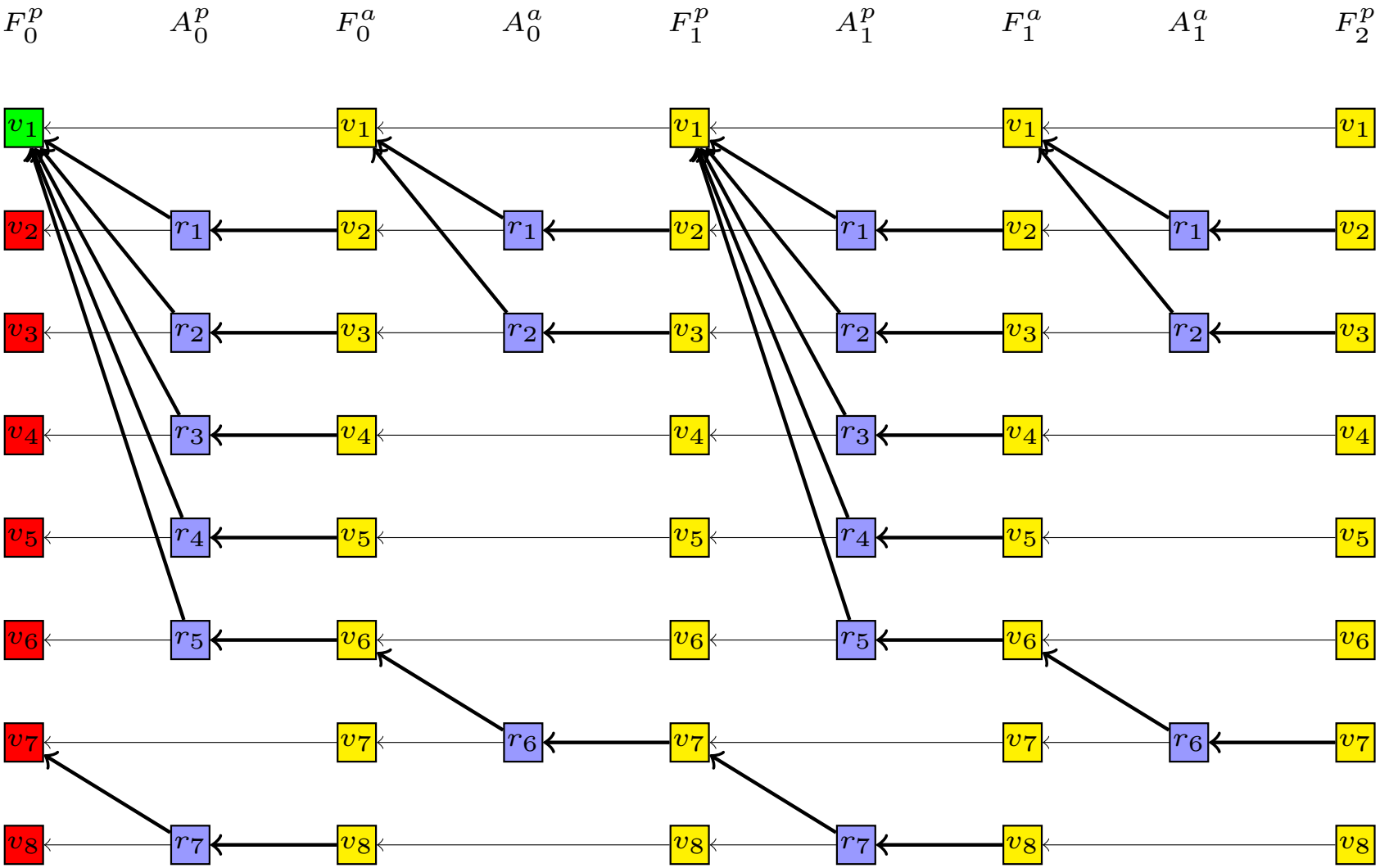
$$\begin{aligned} r_i &= \langle v_1 \rightarrow v_{i+1} \rangle, \quad i = 1, 2, 3, 4, 5 \\ r_j &= \langle v_j \rightarrow v_{j+1} \rangle, \quad j = 6, 7 \end{aligned}$$

- Rules controlled by protagonist: $\{r_1, r_2, r_3, r_4, r_5, r_7\}$
- Rules controlled by antagonist: $\{r_1, r_2, r_6\}$
- Current state: $\{v_1\}$
- Goal: $\{v_1, \ldots, v_8\}$

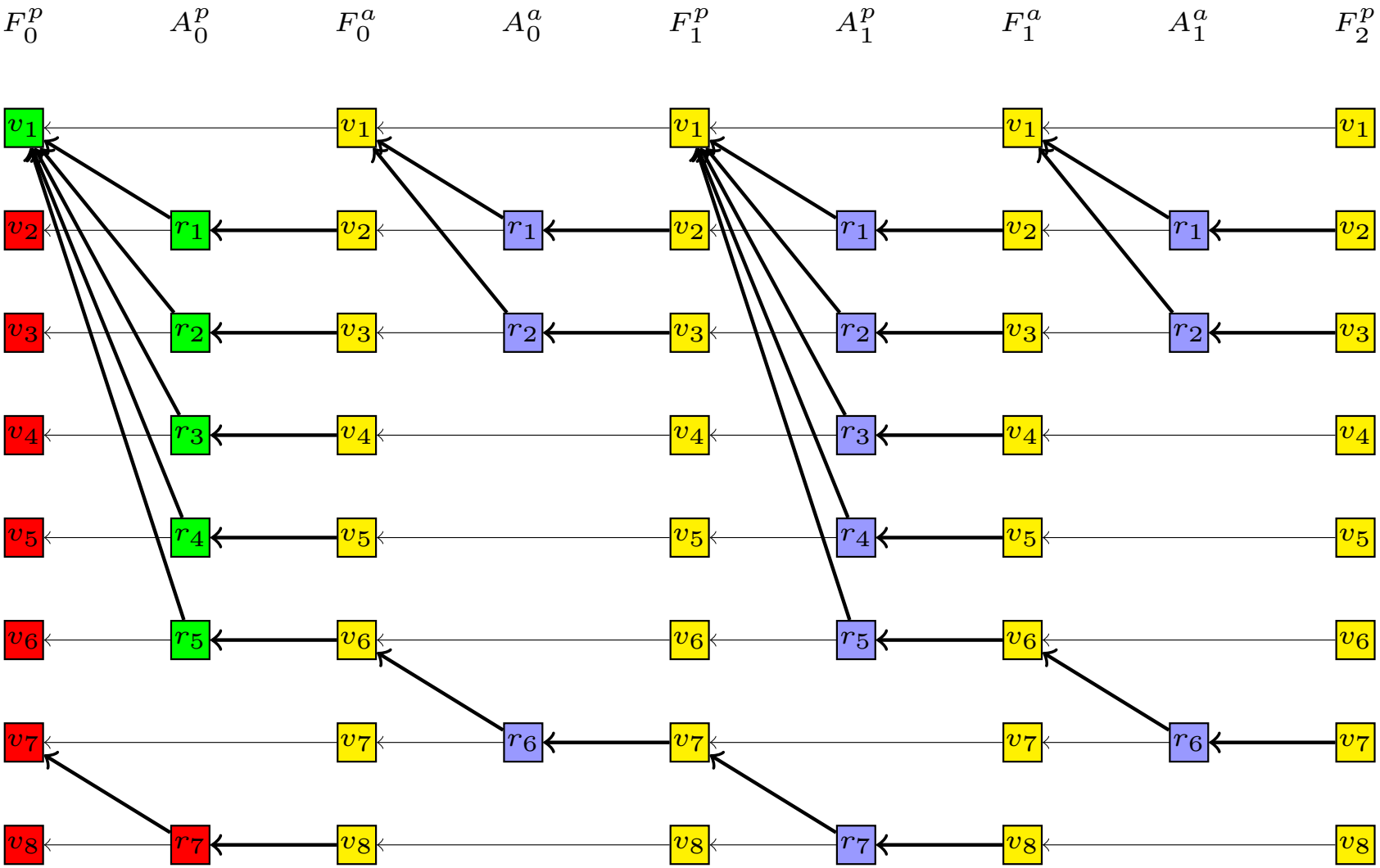
# Heuristic: Relaxed Planning Graph



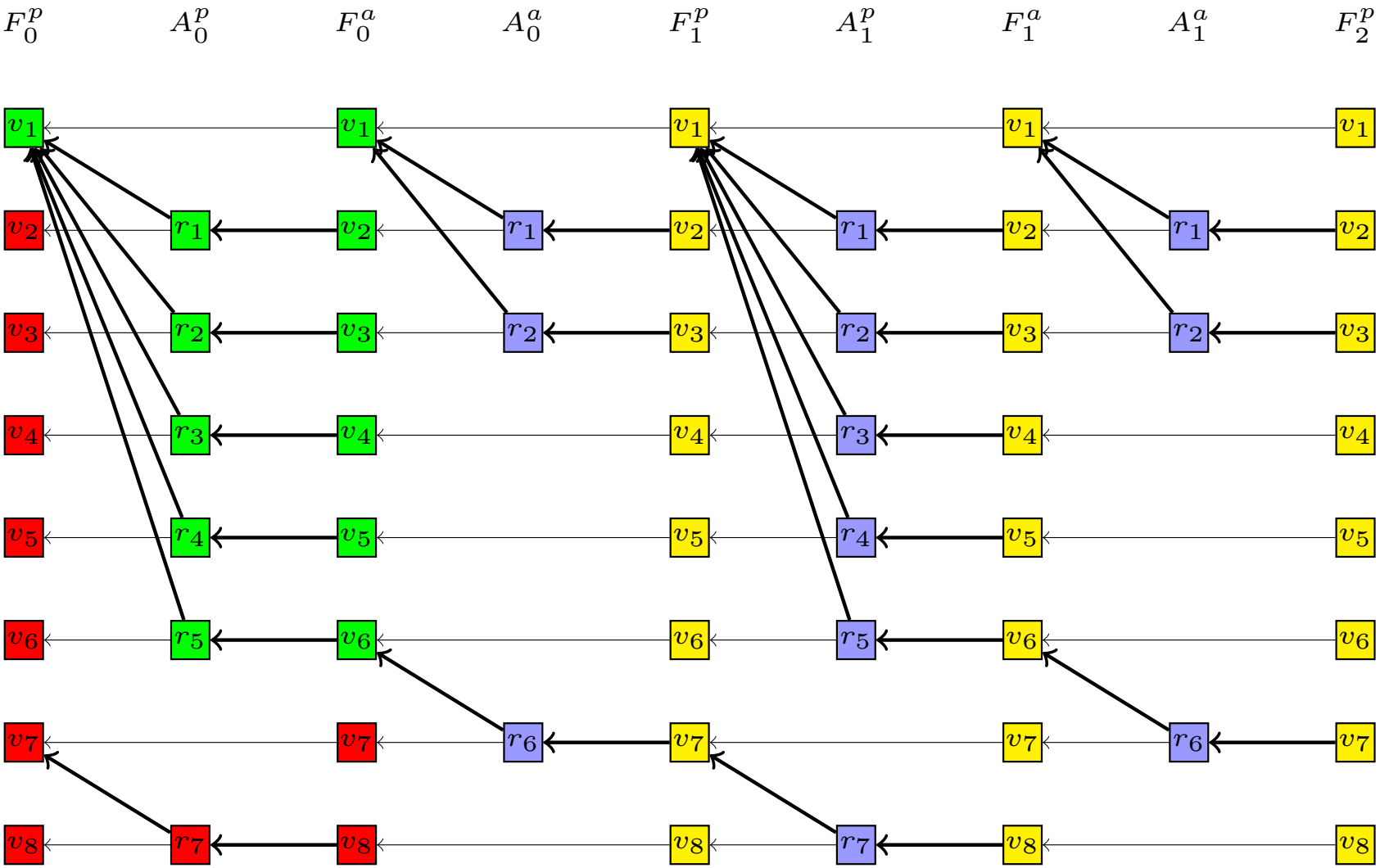
# Heuristic: Relaxed Planning Graph



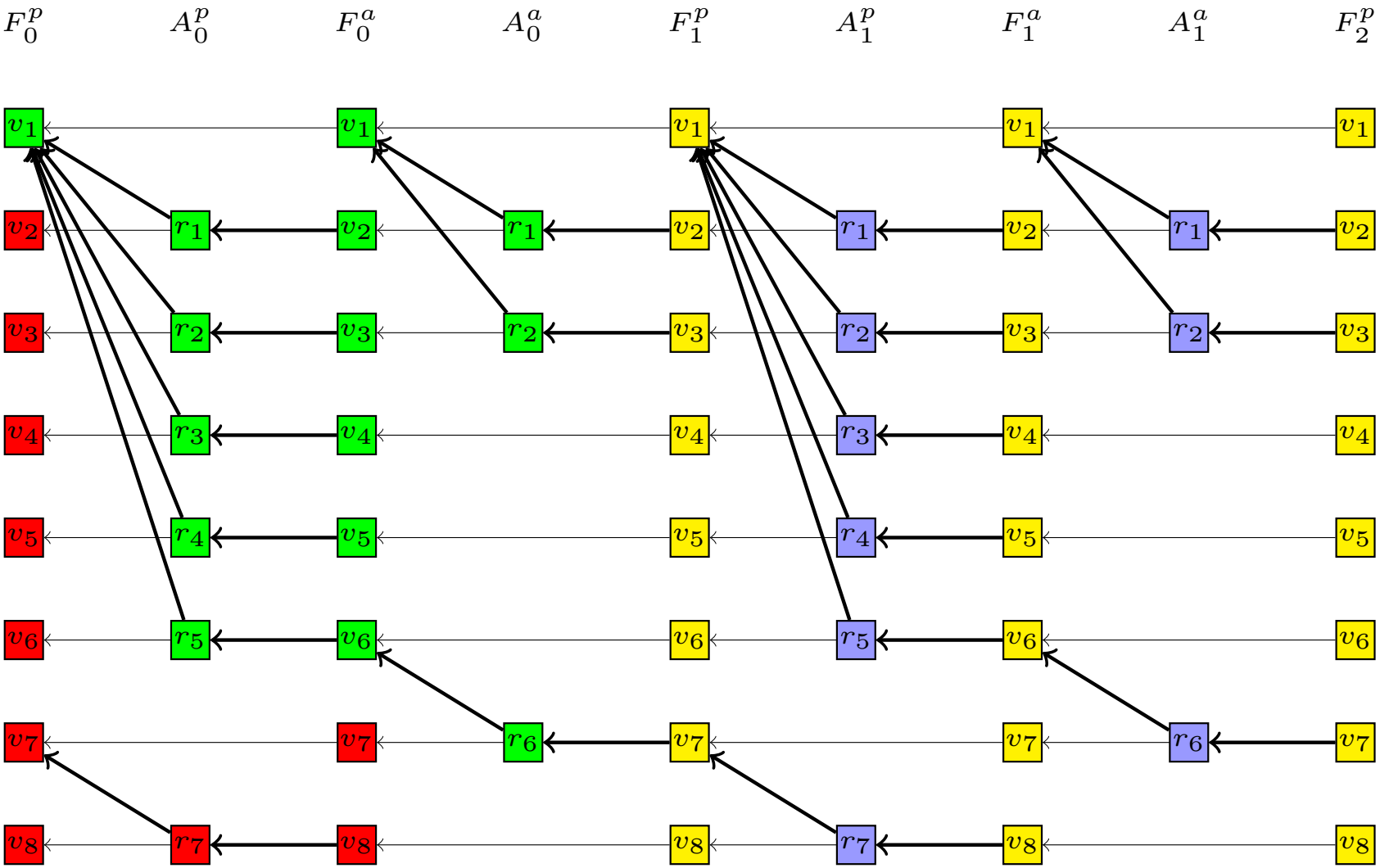
# Heuristic: Relaxed Planning Graph



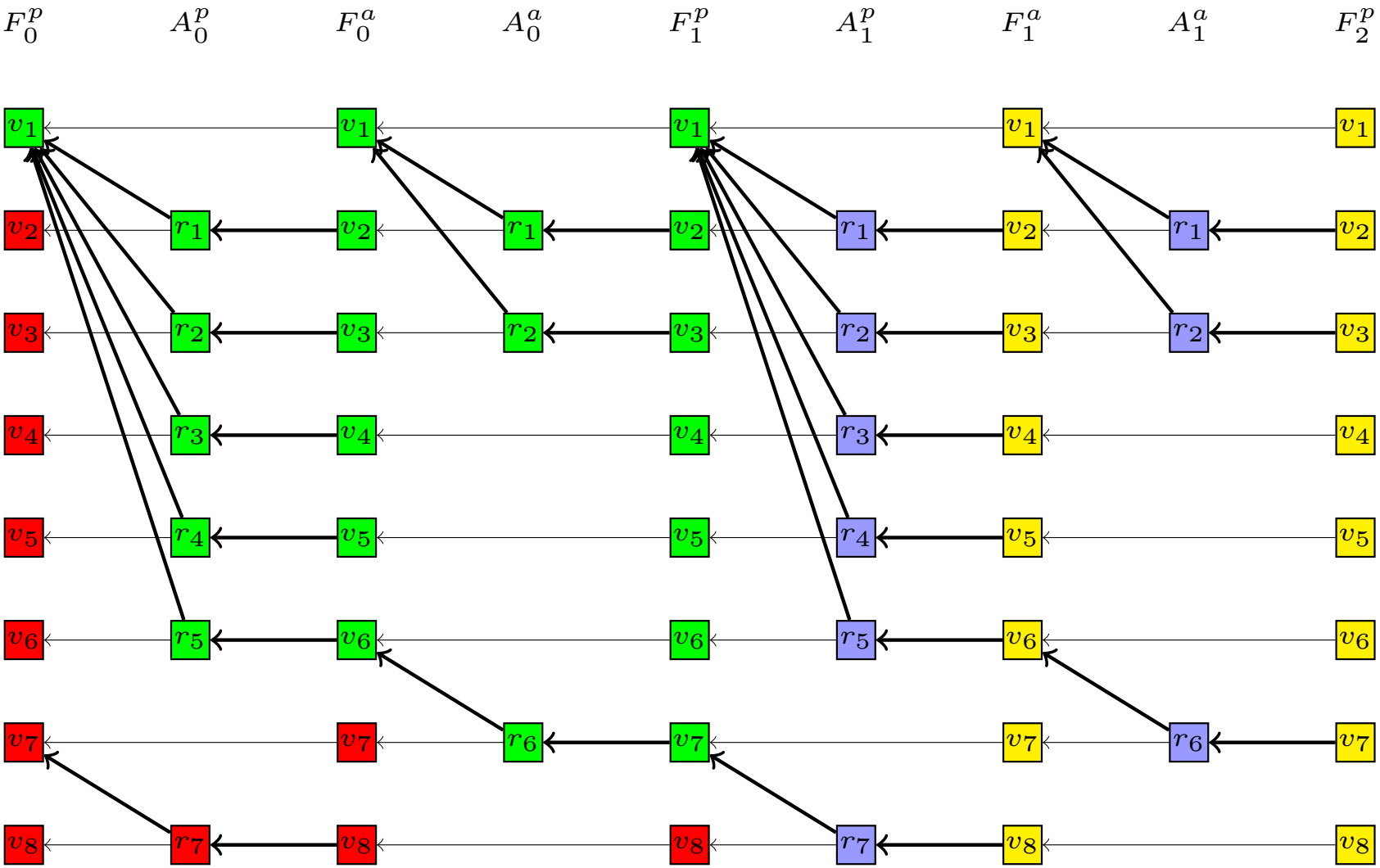
# Heuristic: Relaxed Planning Graph



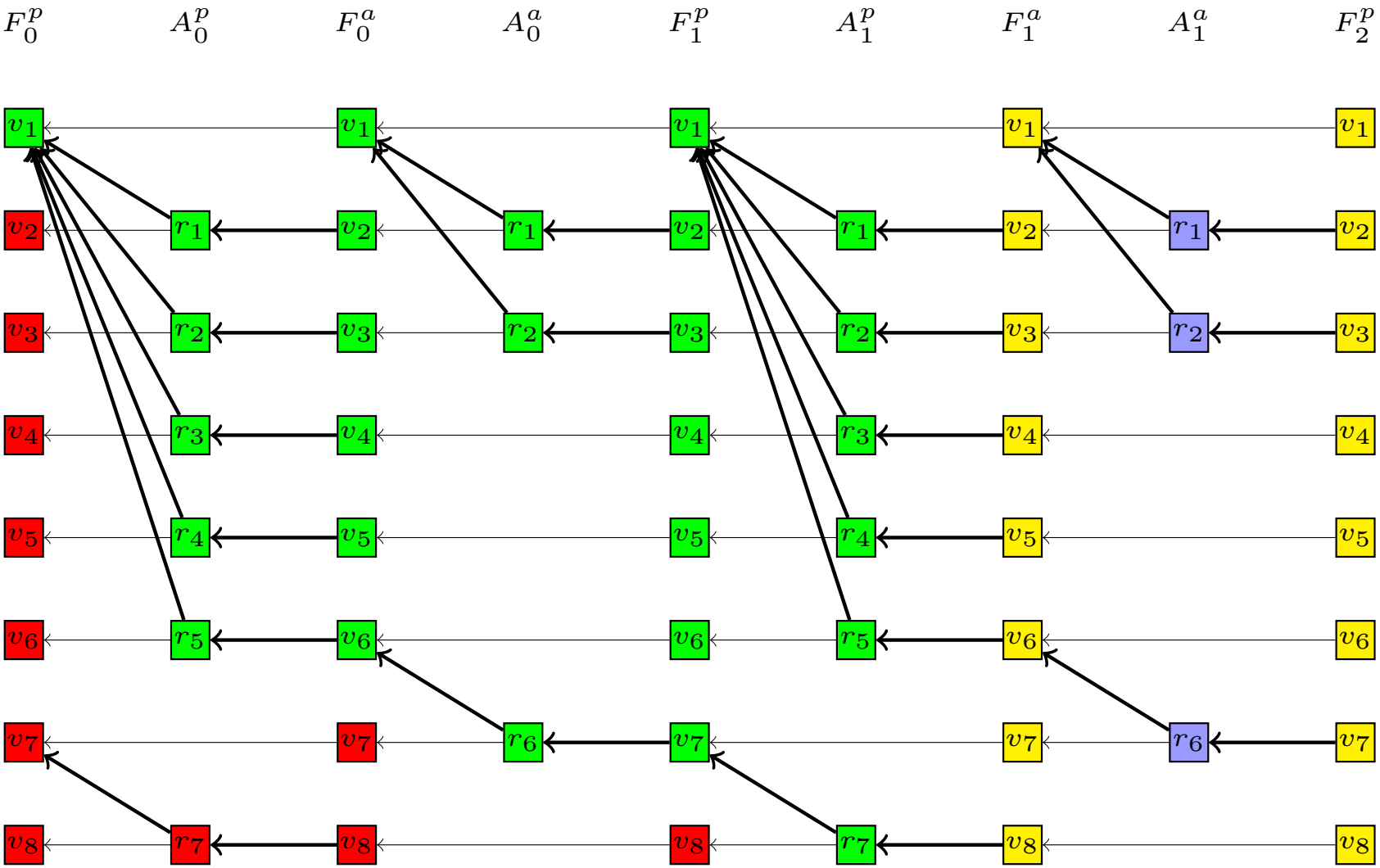
# Heuristic: Relaxed Planning Graph



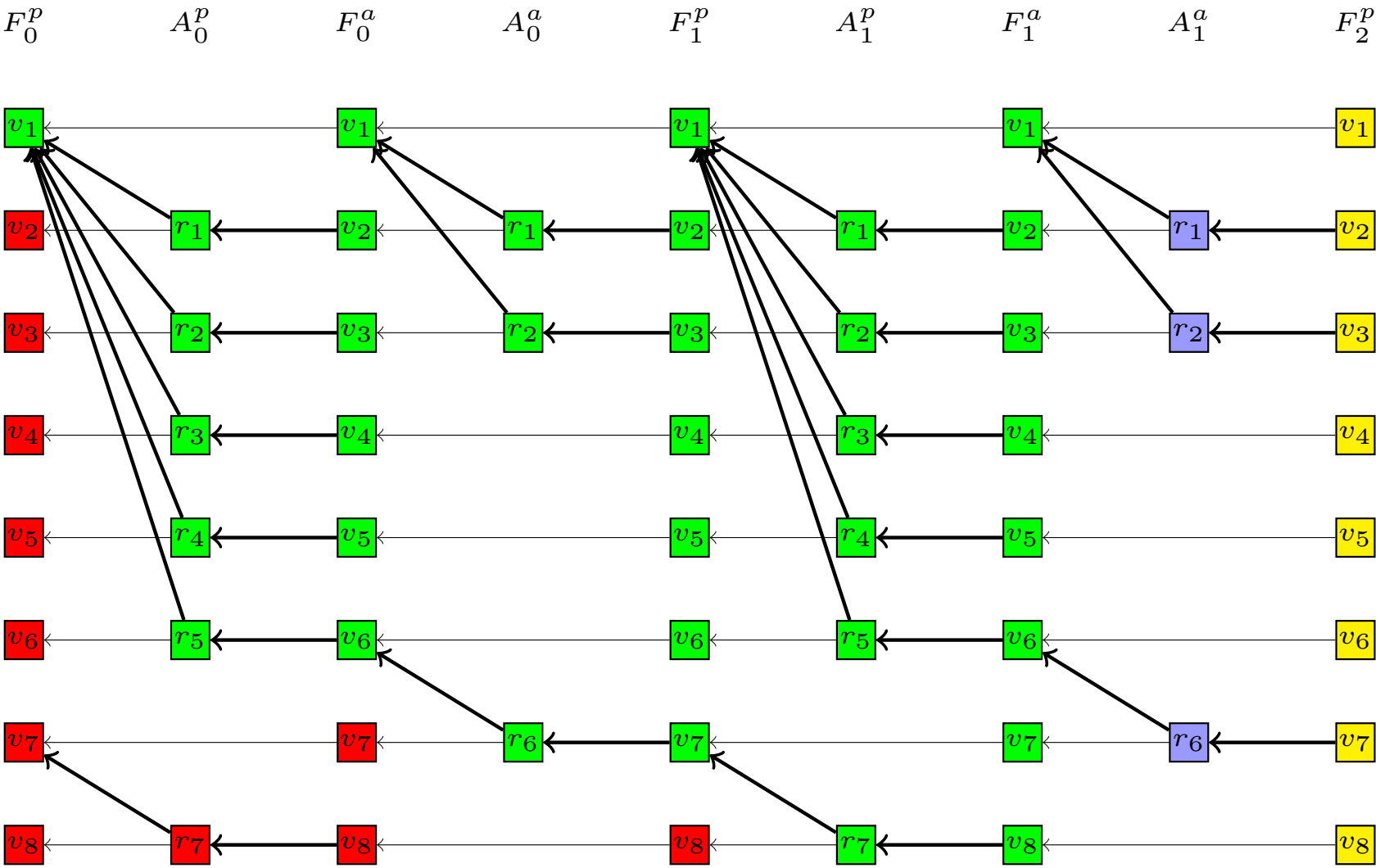
# Heuristic: Relaxed Planning Graph



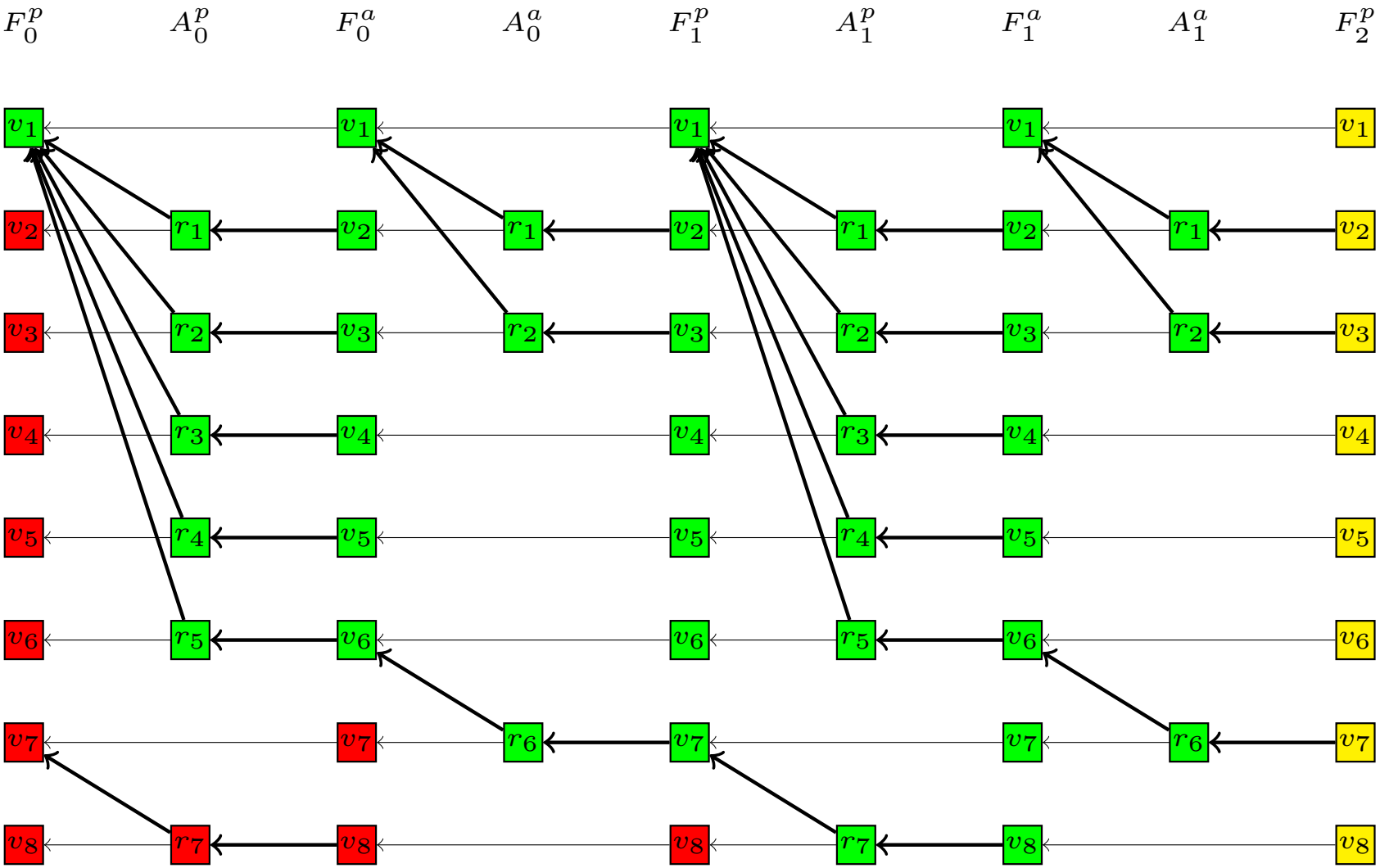
# Heuristic: Relaxed Planning Graph



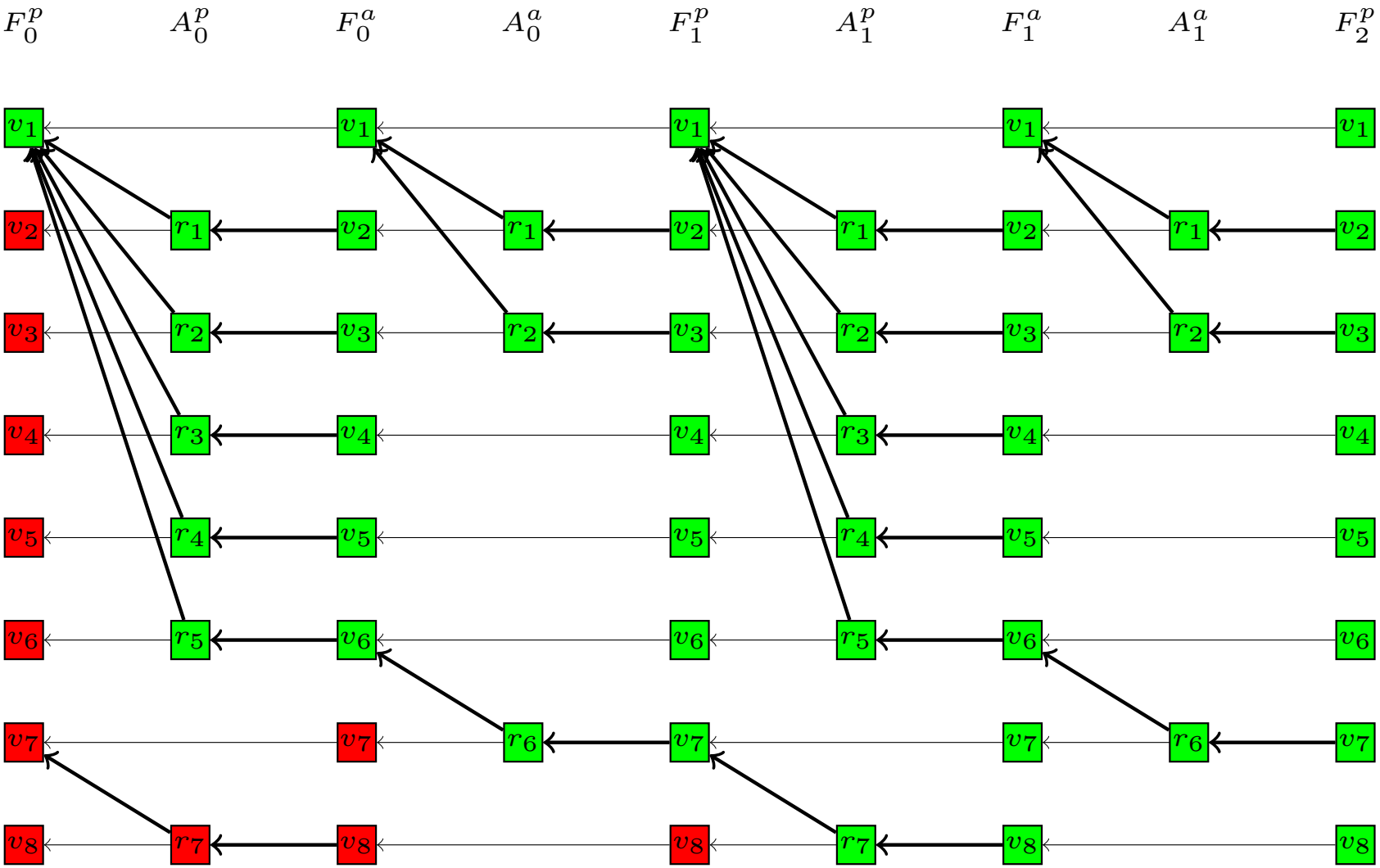
# Heuristic: Relaxed Planning Graph



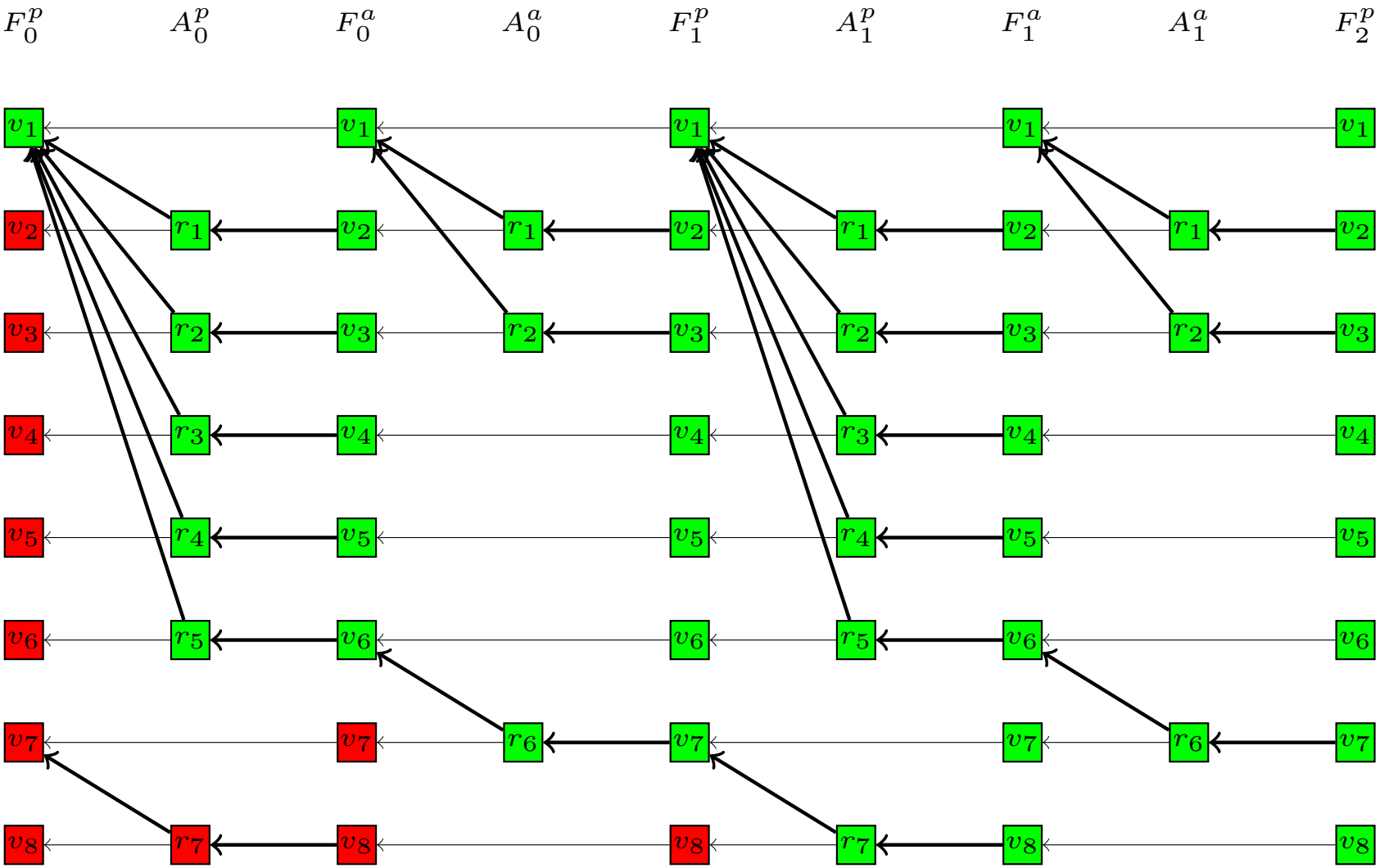
# Heuristic: Relaxed Planning Graph



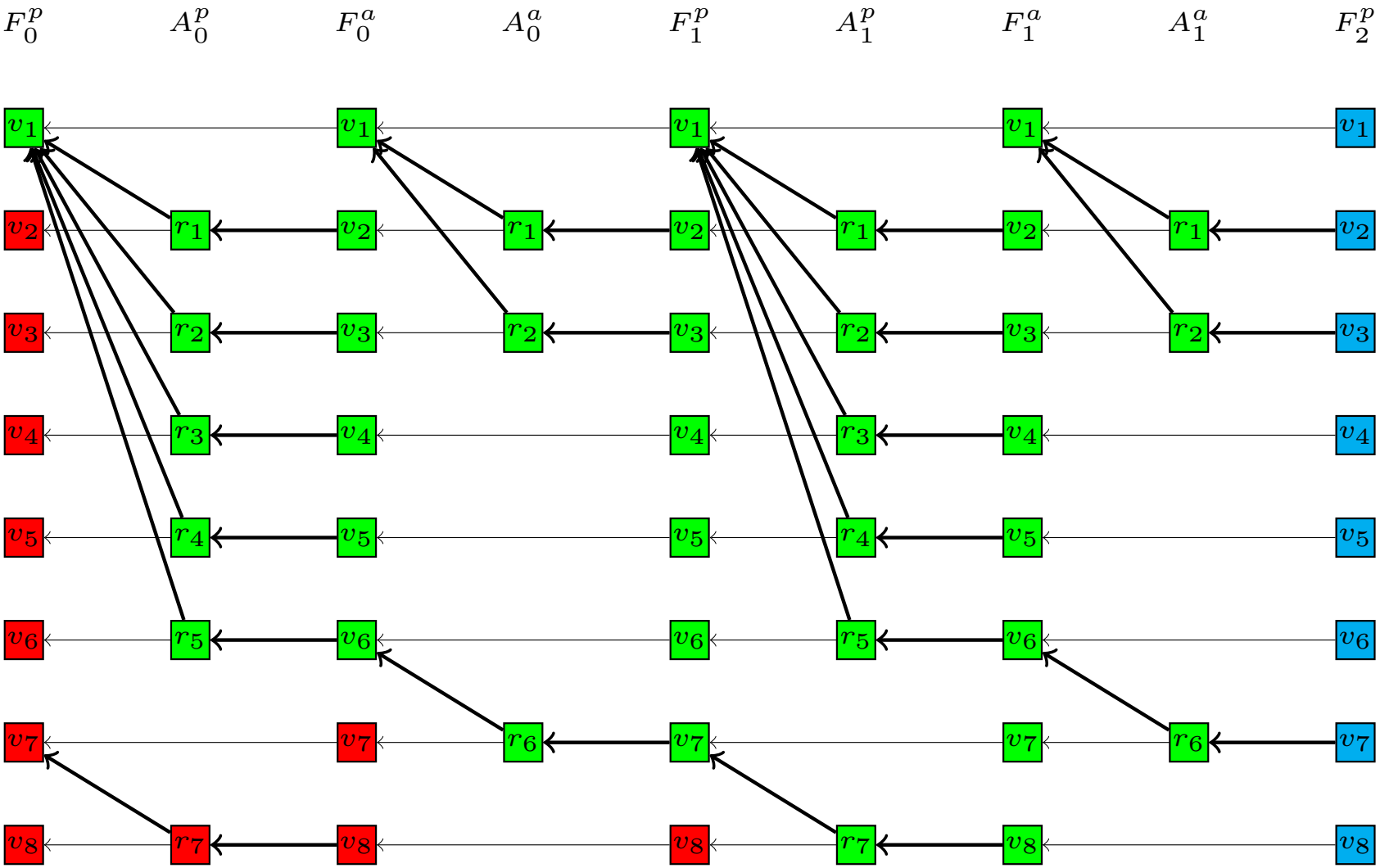
# Heuristic: Relaxed Planning Graph



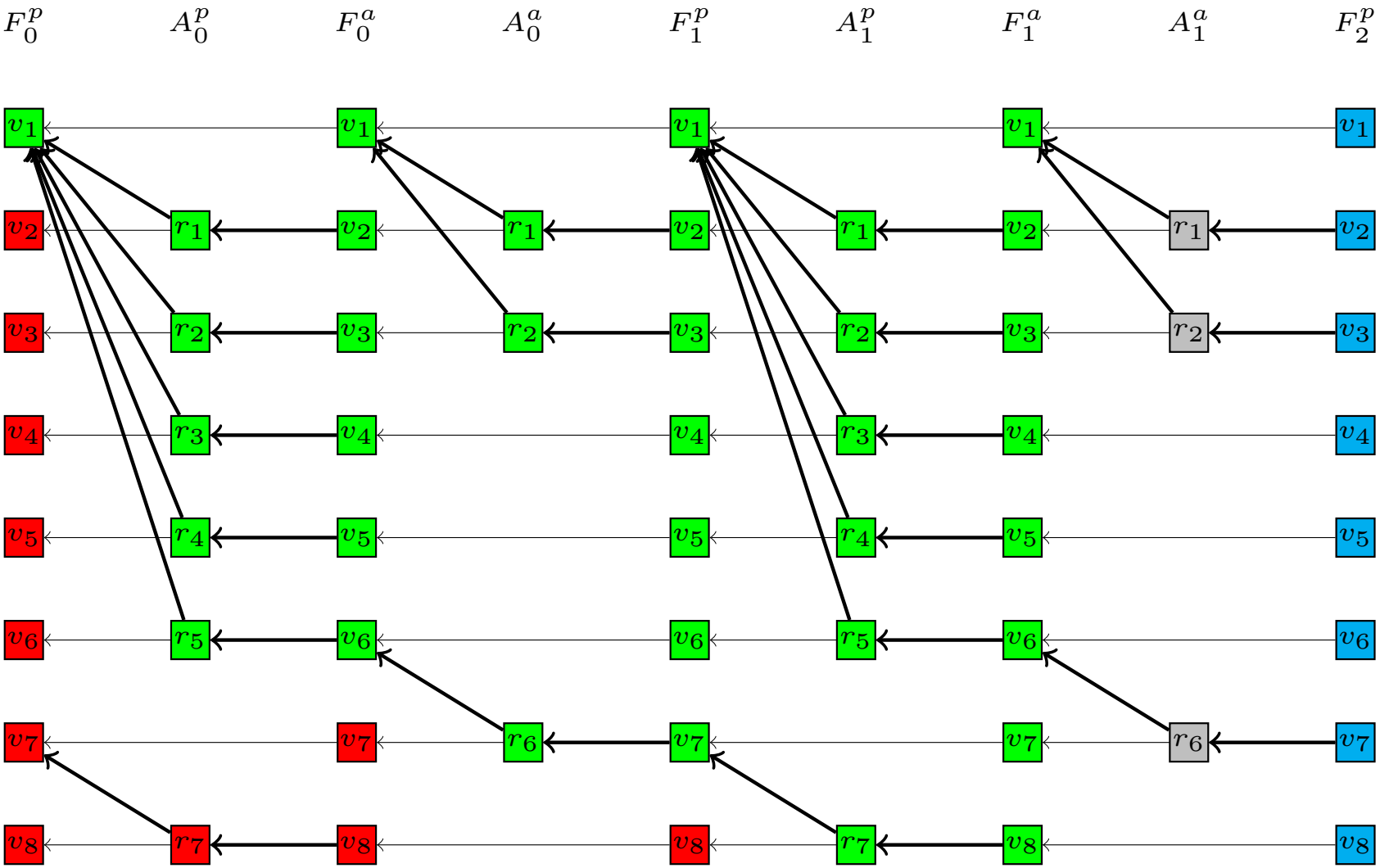
# Heuristic: Relaxed Plan Extraction



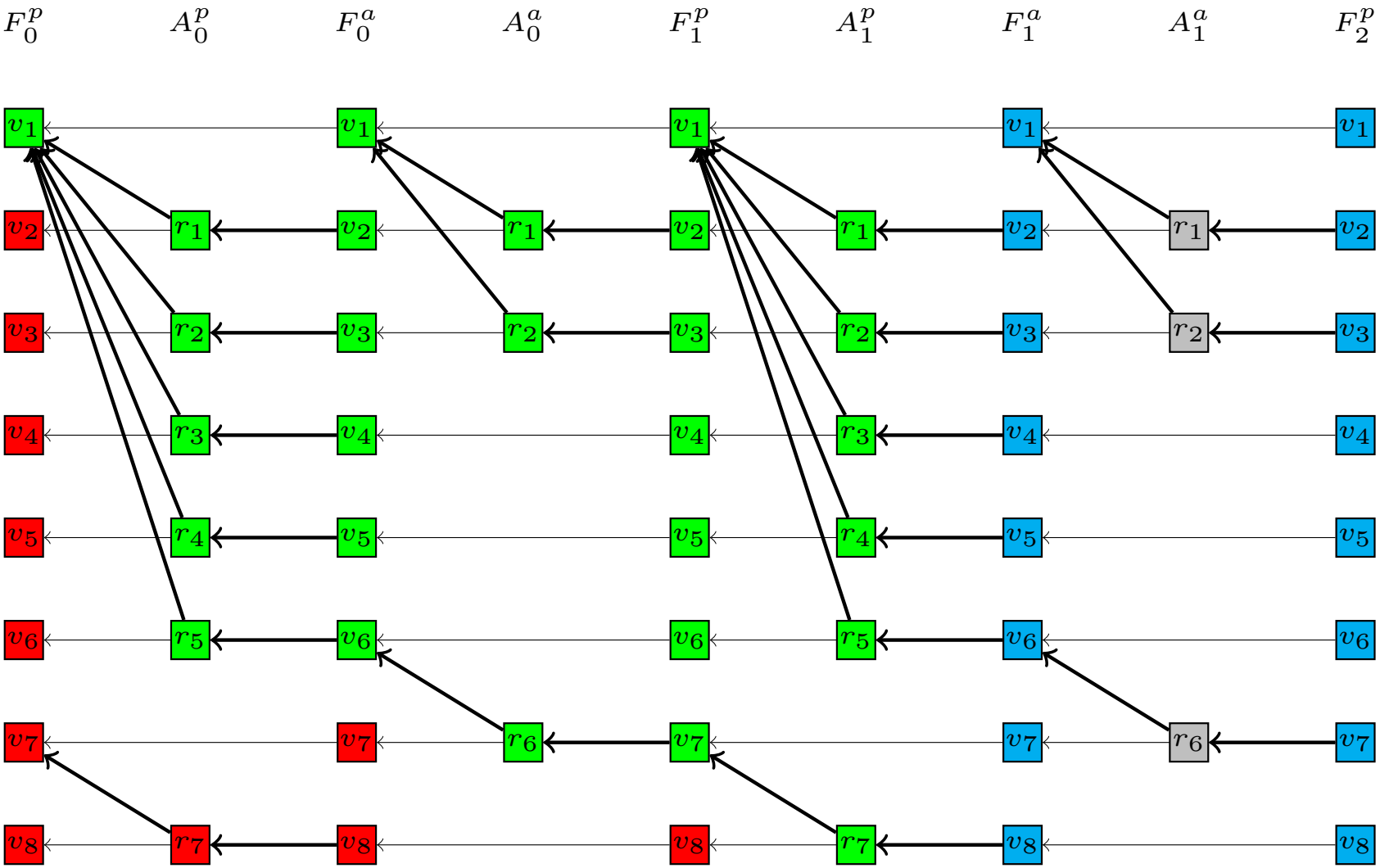
# Heuristic: Relaxed Plan Extraction



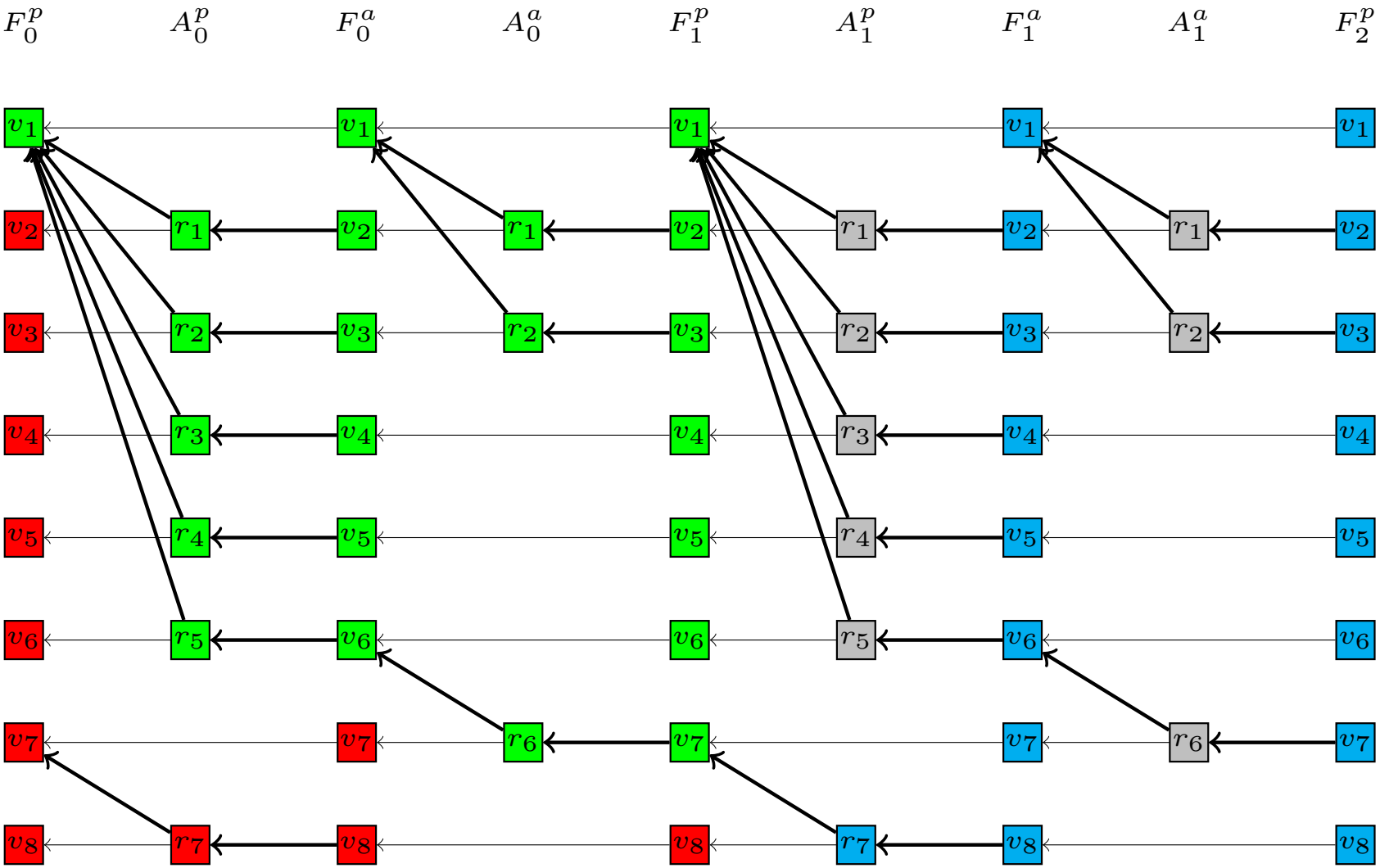
# Heuristic: Relaxed Plan Extraction



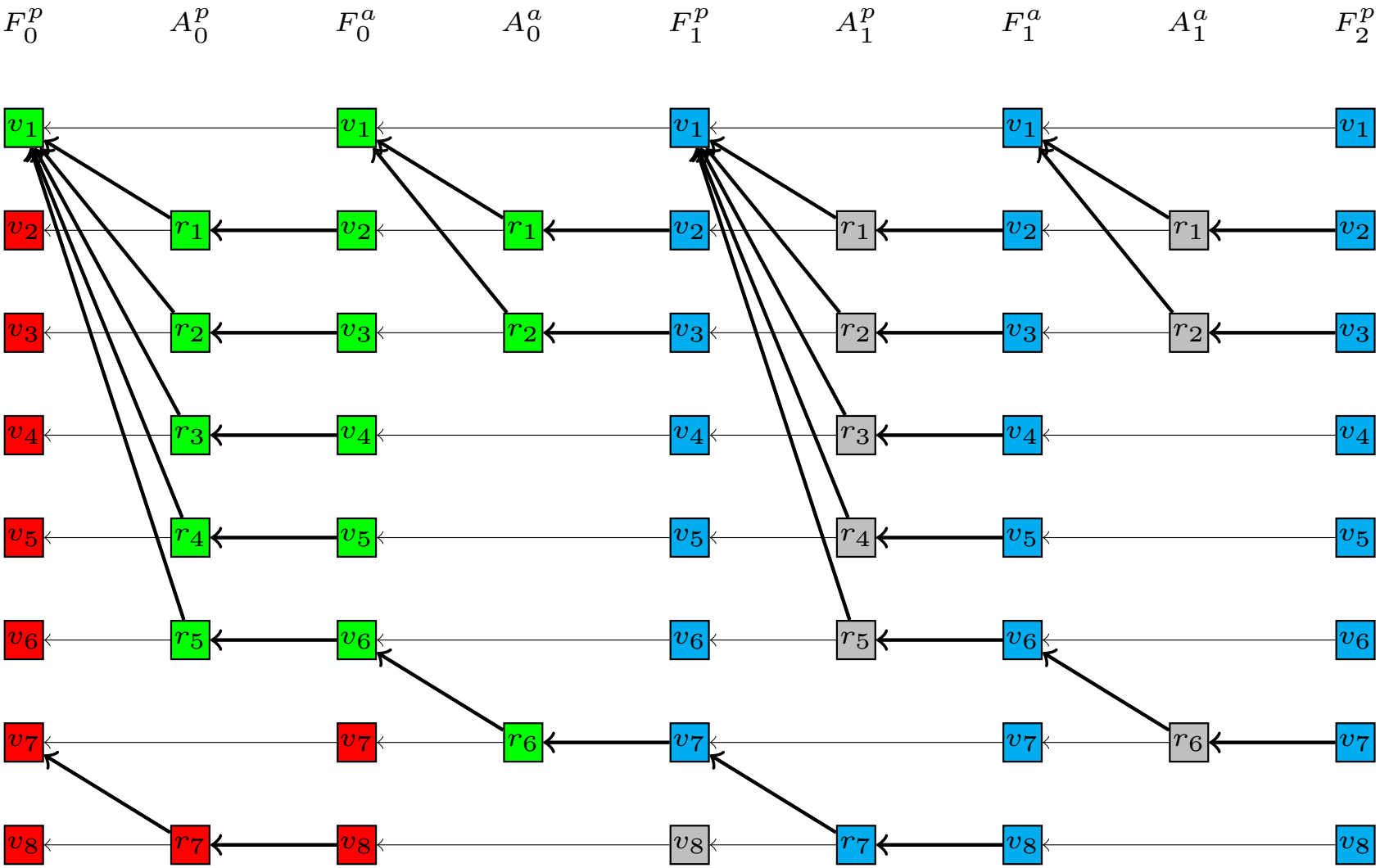
# Heuristic: Relaxed Plan Extraction



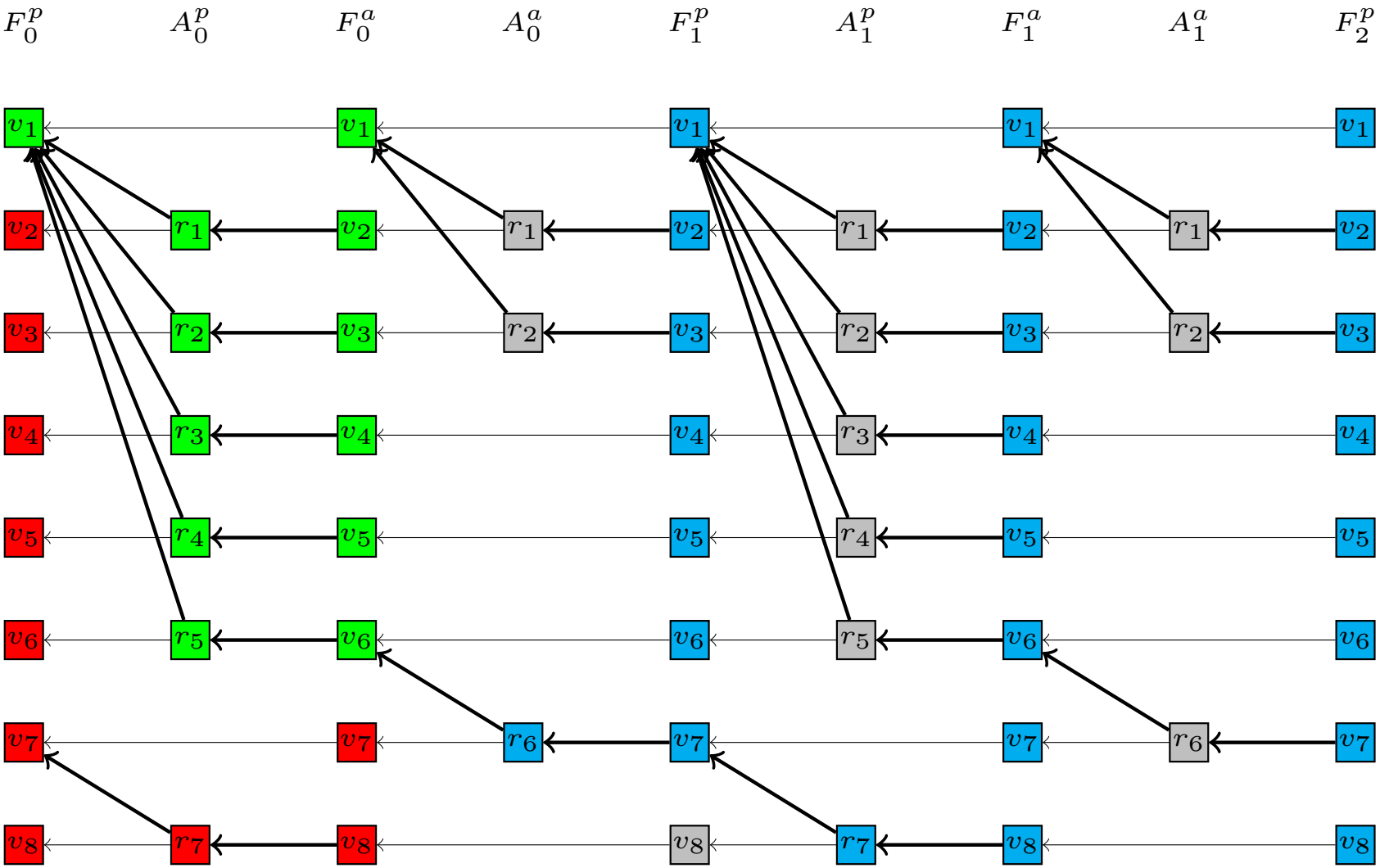
# Heuristic: Relaxed Plan Extraction



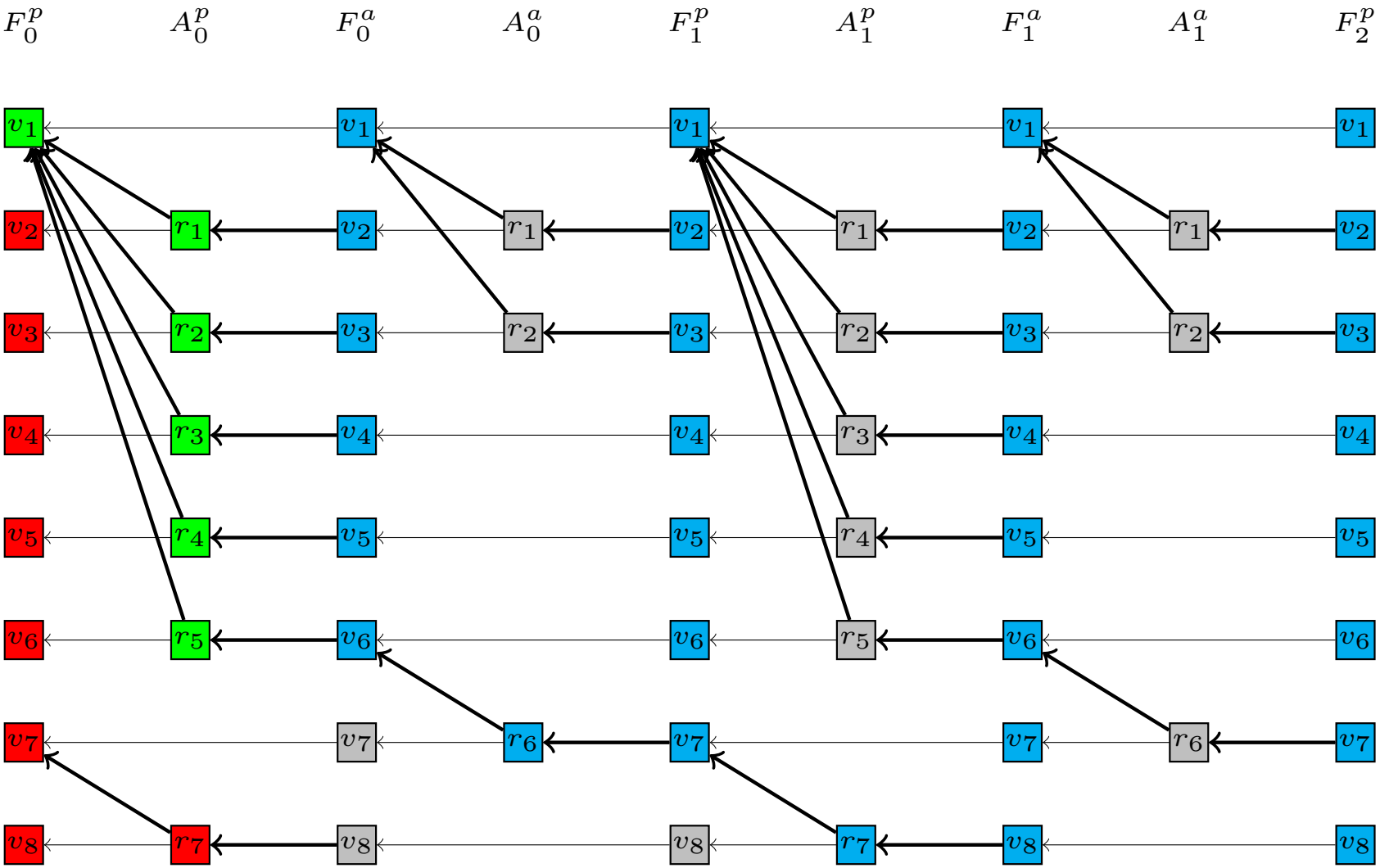
# Heuristic: Relaxed Plan Extraction



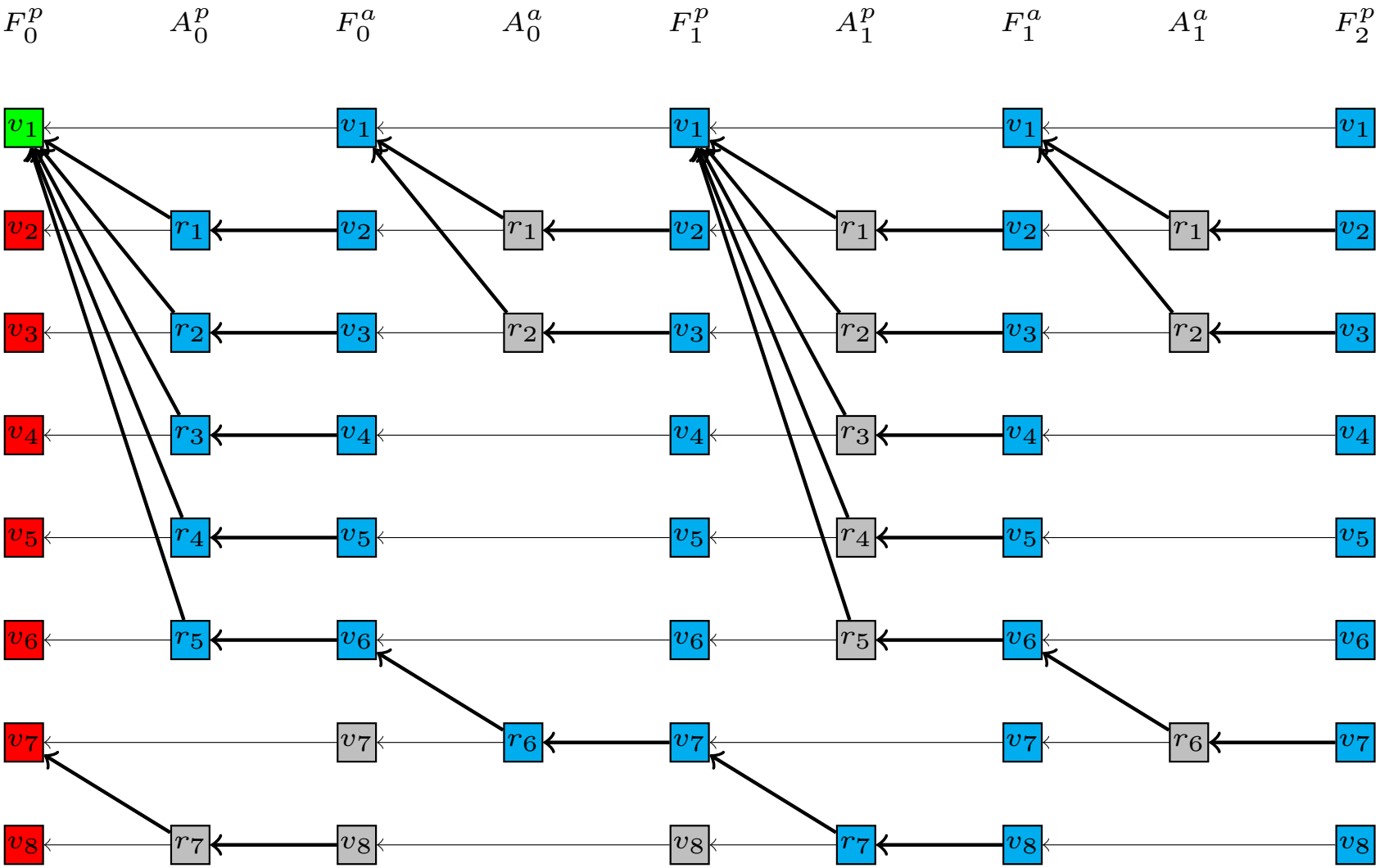
# Heuristic: Relaxed Plan Extraction



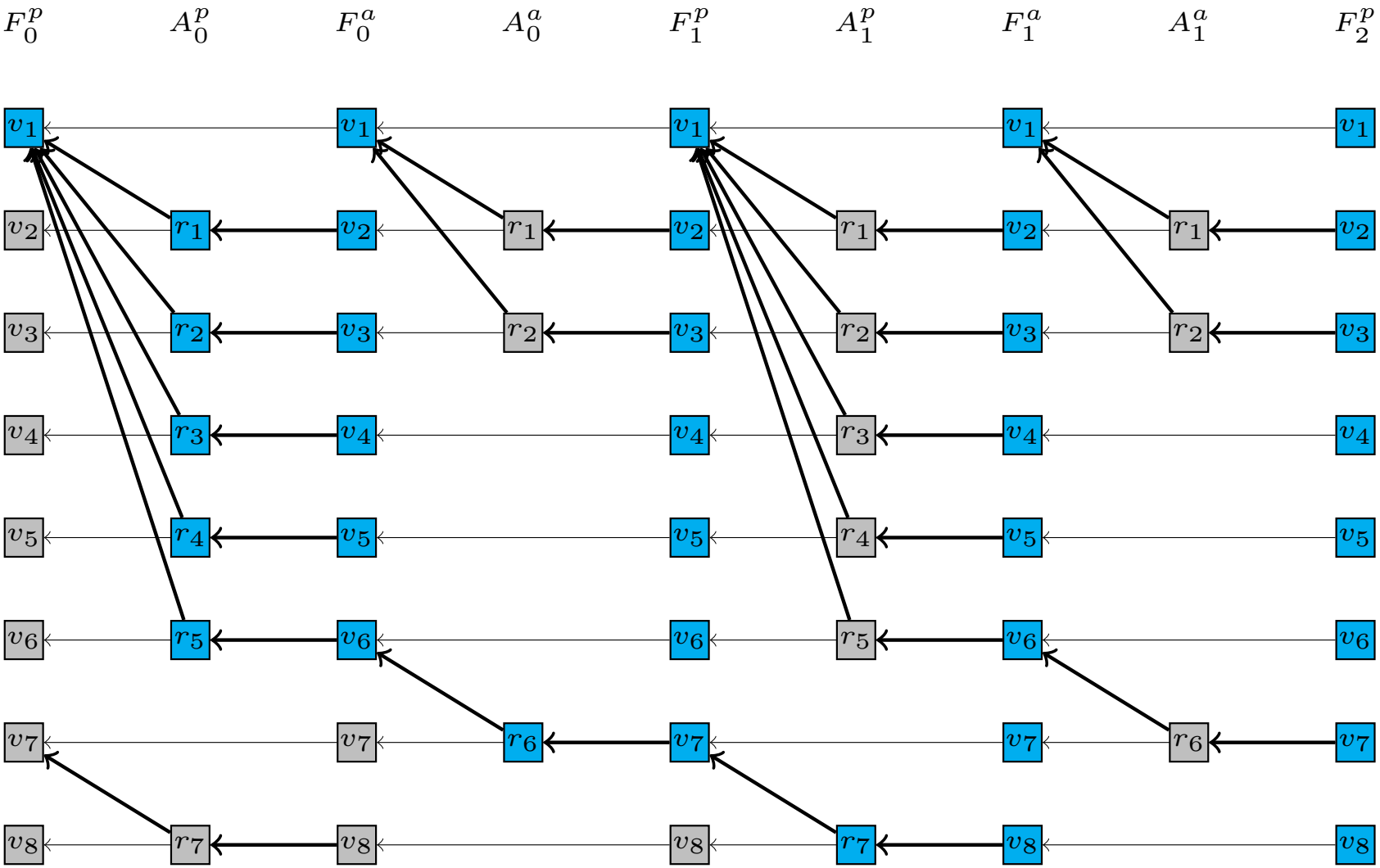
# Heuristic: Relaxed Plan Extraction



# Heuristic: Relaxed Plan Extraction



# Heuristic: Relaxed Plan Extraction



# Heuristic: Relaxed Plan Postprocessing

- Selected rules for protagonist: $\{r_1, r_2, r_3, r_4, r_5, r_7\}$
- Selected rules for antagonist: $\{r_6\}$
- Redistribution of rules:
  - Revised selected rules for protagonist: $\{r_3, r_4, r_5, r_7\}$
  - Revised selected rules for antagonist: $\{r_1, r_2, r_6\}$
- Return heuristic value $2 \cdot |\{r_3, r_4, r_5, r_7\}| = 8$.

# Experiments and Results

- Logistics-like problems as in the example, varying problem sizes
- Comparison of breadth-first search, AO* search with FF heuristic and adversarial FF heuristic, and MBP.

| | | BFS | | AO* + $h_{\mathrm{FF}}$ | | AO* + $h_{\mathrm{adv.\text{-}FF}}$ | | MBP | |
|---|---|---|---|---|---|---|---|---|---|
| $\ell$ | $p$ | time | nodes | time | nodes | time | nodes | time | BDD |
| 2 | 1 | 0.014 | **44** | 0.025 | **37** | **0.026** | **37** | **0.000** | 6601 |
| 2 | 2 | 0.048 | **152** | 0.071 | 88 | **0.072** | **78** | **0.016** | 84424 |
| 3 | 3 | 0.354 | **2106** | **0.202** | **625** | 0.260 | 628 | **0.380** | 23068 |
| 3 | 4 | 0.870 | **8211** | 0.463 | 1871 | **0.232** | **605** | **1.780** | 165718 |
| 3 | 5 | 5.556 | **43785** | 1.437 | 6917 | **0.321** | **794** | **9.041** | 365272 |
| 3 | 6 | **87.691** | **237264** | 16.323 | 63498 | **1.157** | **4164** | 44.287 | 546666 |
| 4 | 6 | **—** | **722750** | **76.718** | **169349** | 82.701 | 194304 | 130.064 | 834704 |
| 4 | 7 | **—** | **771629** | 373.553 | 510738 | **99.639** | **225544** | **—** | — |

$\ell$ : #locations, $p$ : #packages, BDD: #BDD nodes, **red:** worst, **blue:** best

# Conclusion

- Domain-independent heuristics promising approach to conditional/adevrsarial planning
- Explicit-state progression competitive with symbolic regression
- Potential application in General Game Playing
- Future work: Assesment of other domain-independent heuristics in conditional setting