1 "Rough" Claim: Necessary and Sufficient Conditions

We provide necessary and sufficient conditions for the optimal value function V^* to decompose as a linear combination of the features used in a generalized and optimal policy for the problem. As we will see, the conditions are expressed as a system of linear equations, one for each policy rule in the generalized policy. Later, we provide sufficient conditions that guarantee the decomposition.

Features are state functions, either non-negative or Boolean valued. A generalized policy is a set of policy rules of the form $C \mapsto E$, where C is a set of Boolean conditions (either p or $\neg p$ for Boolean features, or $n > 0$ or $n = 0$ for numerical features), and E is a set of effects (either, p or $\neg p$ for Boolean features, or $n\downarrow$ or $n\uparrow$ for numerical features). A policy rule $C \mapsto E$ is compatible with a state transition (s, s') iff s satisfies C and E satisfies the change of feature values $(f(s), f(s'))$ across the transition, where $f(s)$ denotes the value of the features at state s. A state trajectory $s_0, s_1, \ldots, s_n$ is compatible with a policy π if each transition (s_i, s_{i+1}) is compatible with a rule in π, and it's maximal if s_n is a goal state, or s_n has no successor s' such that (s_n, s') is compatible with π. The policy π solves a problem if for any state s in the problem, all the maximal state trajectories compatible with π reach the goal. It is optimal iff the goal-reaching state trajectories compatible with it are of minimum length.

We assume throughout a fixed generalized policy π defined over a set Φ of features, and that the policy is optimal. General features in Φ are denoted by ϕ while numerical (resp. Boolean) features are denoted by x (resp. p). We assume without loss of generality that every policy rule that modifies a Boolean feature has as precondition the negation of the feature. This condition can always be met by replacing offending actions with one or more equivalent actions. Finally, we assume that the decrements of a numerical feature always affect the value of the feature by exactly one unit.

A linear decomposition of V^* is an expression of the form $V^* = \sum_{\phi\in\Phi} w_\phi\, \phi$ where w_ϕ is the weight assigned to feature ϕ, and for Boolean features ϕ, $\phi(s)$ is 0 (resp. 1) whether ϕ is false (resp. true) at s. For given state s, the linear decomposition of V^* translates into $V^*(s) = \sum_{\phi\in\Phi} w_\phi\, \phi(s)$.

The following results establishes a necessary condition for linear decomposition of the optimal value function. As we will see, additional mild assumptions guarantee that this condition is also sufficient. The condition can be checked by solving a system of linear equations obtained solely from the policy rules in π, without reference to any concrete problem P. It is still open to find further conditions that imply it.

Theorem 1 (Necessity)**.** *Let π be an optimal policy with no superfluous policy rules (i.e., the removal of any rule from π yields a policy that is incomplete). For any numerical feature x, let us assume that the increments of x are always by the* ***constant amount*** *Δ_x. If V^* is linearly decomposed, the following system of linear equations on the variables $\{w_\phi : \phi \in \Phi\}$ has solution: for each policy rule $a : C \mapsto E$:*

$$1 \;=\; \sum_{x\in\Phi} w_x \Big([\![x\downarrow \in E]\!] - \Delta_x [\![x\uparrow \in E]\!] \Big) + \sum_{p\in\Phi} w_p \Big([\![\neg p \in E]\!] - [\![p \in E]\!] \Big). \tag{1}$$

In this expression, x (resp. p) ranges over the numerical (resp. Boolean) features in Φ.

Proof. Let s be a non-goal state in a problem P, and let s' be a successor of s such that the transition (s, s') is compatible with π. Then, for all such transitions (s, s') and the policy rules $a : C \mapsto E$ in π that are compatible with (s, s'):

$$\begin{aligned}
1 \;&=\; V^*(s) - V^*(s') && (2)\\
&=\; \sum_{\phi\in\Phi} w_\phi(\phi(s) - \phi(s')) && (3)\\
&=\; \sum_{x\in\Phi} w_x(x(s) - x(s')) + \sum_{p\in\Phi} w_p(p(s) - p(s')) && (4)\\
&=\; \sum_{x\in\Phi} w_x \Big([\![x\downarrow \in E]\!] - \Delta_x [\![x\uparrow \in E]\!] \Big) + \sum_{p\in\Phi} w_p \Big([\![\neg p \in E]\!] - [\![p \in E]\!] \Big) && (5)
\end{aligned}$$

where the last equality is by the assumption on the preconditions of actions, and $x(s) - x(s')$ is 1 (resp. $-\Delta_x$ or 0) whether the effect E decrements (resp. increments or doesn't affect) x. □

Lemma 2. *Let $s_0, a_0, s_1, a_1, \ldots, s_n$ be a trajectory in state state compatible with policy π, where each $a_i : C_i \mapsto E_i$ is a policy rule in π. For any Boolean feature p,*

$$\sum_{i=0}^{n-1} \Big([\![\neg p \in E_i]\!] - [\![p \in E_i]\!] \Big) \;=\; p(s_0) - p(s_n)\,. \tag{6}$$

Similarly, under the assumption that the increments of the numerical feature x are always by the constant amount Δ_x, for any numerical feature x:

$$\sum_{i=0}^{n-1} \Big([\![x{\downarrow} \in E_i]\!] - \Delta_x [\![x{\uparrow} \in E_i]\!] \Big) \;=\; x(s_0) - x(s_n)\,. \tag{7}$$

Proof. For the first claim, we consider 4 cases that result of combining the truth value of the feature p at the states s_0 and s_n. For the case $p(s_0) = p(s_n) = 1$, the sequence of effects on p along the trajectory must be of the form $(-, +, \ldots, -, +)$ where $-$ denotes the effect $\neg p$ and p denotes the effect p. Each $-/+$ contributes 1/-1 to the sum for a total net sum of zero, which equals the right-hand side of (6). If $p(s_0) = 1 - p(s_n) = 1$, the sequence of effects is of the form $(-, +, \ldots, -)$ that results in a value of 1, which again equals the right-hand side. The other two cases are similar.

For the second claim, the difference in value for x between the first and last states in the trajectory is equal to the net effect of the increments and decrements. □

Definition 3 (Boolean Valuation and Policy Graph)**.** *A Boolean valuation for a set of features Φ is joint valuation for the Boolean features in Φ together with the Boolean value of the literals $x = 0$ for the numerical features in x. The policy graph for policy π over features Φ is a directed graph whose vertices are the Boolean valuations for Φ, and there is an edge (σ, σ') between two Boolean valuations if the transition (σ, σ') is compatible with some policy rule in π.*

Theorem 4. *Assume that the policy graph has a unique goal vertex where all numerical features are equal to zero, and that the increments of the numerical feature x are always by the constant amount Δ_x. V^* is linearly decomposed iff the system of equations* (1) *holds.*

Proof. Necessity is established in Theorem 1. Thus, let us assume that the system of equations has solution given by weights $\{w_\phi\}_{\phi \in \Phi}$. We need to show that V^* is linearly decomposed. Let $s_0, a_0, s_1, a_1, \ldots, s_n$ be a trajectory compatible with π seeded at $s_0 = s$ that reaches the goal state s_n. We have,

$$\begin{aligned}
V^*(s) - V^*(s_n) \;&=\; \sum_{i=0}^{n-1} V^*(s_i) - V^*(s_{i+1}) && (8)\\
&=\; \sum_{i=0}^{n-1} \Big(\sum_x w_x \Big([\![x{\downarrow} \in E_i]\!] - \Delta_x [\![x{\uparrow} \in E_i]\!] \Big) + \sum_p w_p \Big([\![\neg p \in E_i]\!] - [\![p \in E_i]\!] \Big) \Big) && (9)\\
&=\; \sum_x w_x \sum_{i=0}^{n-1} \Big([\![x{\downarrow} \in E_i]\!] - \Delta_x [\![x{\uparrow} \in E_i]\!] \Big) + \sum_p w_p \sum_{i=0}^{n-1} \Big([\![\neg p \in E_i]\!] - [\![p \in E_i]\!] \Big) && (10)\\
&=\; \sum_x w_x \Big(x(s) - x(s_n) \Big) + \sum_p w_p \Big(p(s) - p(s_n) \Big) && (11)\\
&=\; \sum_x w_x\, x(s) + \sum_p w_p\, p(s) + \text{const.} && (12)
\end{aligned}$$

where the fourth equality follows by Lemma 2, and the last by the assumption of unique goal vertex in the policy graph and that all numerical features equal 0 at goal. The constant is given by the value of the Boolean features at the goal; i.e., $-\sum_p w_p p(s_n)$. The claim follows by noting $V^*(s_n) = 0$. □

1.1 Examples

1.1.1 Blocksworld: $clear(x)$

Optimal policy for clear given by two rules $\{H\} \mapsto \{\neg H\}$ and $\{\neg H, n>0\} \mapsto \{H, n\downarrow\}$ where H denotes the agent holds a block, and n counts the number of blocks above x. The system of equations is $\{1 = w_H, 1 = w_n - w_H\}$ whose solution is clearly $\{w_H = 1, w_n = 2\}$. For the goal of getting $clear(x)$ and empty hand, both features have zero value at goal. Therefore, the optimal value function decomposes as expected:

$$V^*(s) \;=\; 2n(s) + H(s)\,. \tag{13}$$

1.1.2 Blocksworld: $on(x, y)$

Simple case. Let us assume that the blocks x and y are *not in the same tower* initially, and let us consider the feature x (resp. y) that counts the number of blocks above x (resp. y), H whether a block different from x is being held, X whether the block X is being held, and g whether block x is on block y. The goal is to achieve $g \wedge \neg H$. An optimal policy is obtained with the rules $\{\neg H, x>0\} \mapsto \{H, x\downarrow\}$, $\{\neg H, y>0\} \mapsto \{H, y\downarrow\}$, and $\{H\} \mapsto \{\neg H\}$ for clearing the blocks x and y, and $\{\neg H, x=0, y=0\} \mapsto \{H, X\}$ and $\{H, X, y=0\} \mapsto \{\neg H, \neg X, g\}$. The equations are:

$$1 \;=\; w_x - w_H\,, \tag{14}$$
$$1 \;=\; w_y - w_H\,, \tag{15}$$
$$1 \;=\; w_H\,, \tag{16}$$
$$1 \;=\; -w_H - w_X\,, \tag{17}$$
$$1 \;=\; w_H + w_X - w_g\,. \tag{18}$$

The solution is $\{w_H = 1, w_x = 2, w_y = 2, w_X = -2, w_g = -2\}$. At the goal all features have zero value except g that is 1. The optimal value function then decomposes as

$$V^*(s) \;=\; 2x(s) + 2y(s) - 2g(s) - 2X(s) + H(s) + 2\,. \tag{19}$$

For example, in the initial state, $V^*(s) = 2x + 2y + 2$, meaning that the number of steps to do is 2 plus twice the sum of the number of blocks above x and y.

General case. We now consider the general case where both blocks can be in the same tower. The set of features is $\Phi = \{A, B, E, X, Y, D, Z, n, m\}$ where A represents whether x is above y, B whether y is above x, E whether the gripper is empty, X whether x is being held, Y whether y is being held, D whether the goal has been achieved, Z whether holding X and y is clear, n counts the number of blocks above x but different from or not above y, and m counts the number of blocks above y but different from or not above x. That is, if block y is above x, n counts the number of blocks above x that are below y. If not, it counts the blocks that are above x. The policy rules are:

$$\text{Pick-above-}x: \quad \{E, \neg B, n>0\} \mapsto \{\neg E, n\downarrow\}\,, \tag{20}$$
$$\text{Pick-above-}y: \quad \{E, \neg A, m>0\} \mapsto \{\neg E, m\downarrow\}\,, \tag{21}$$
$$\text{Pick-}y\text{-on-}x: \quad \{E, \neg Y, B, m=0\} \mapsto \{\neg E, Y, \neg B\}\,, \tag{22}$$
$$\text{Put-away:} \quad \{\neg E, \neg X, \neg Y\} \mapsto \{E\}\,, \tag{23}$$
$$\text{Put-}y\text{-away:} \quad \{\neg E, Y\} \mapsto \{E, \neg Y\}\,, \tag{24}$$
$$\text{Pick-}x\text{-not-on-}y: \quad \{E, \neg X, A, \neg D, n=0\} \mapsto \{\neg E, X, \neg A\}\,, \tag{25}$$
$$\text{Put-}x\text{-away:} \quad \{\neg E, X\} \mapsto \{E, \neg X\}\,, \tag{26}$$
$$\text{Pick-}x\text{-for-goal:} \quad \{E, \neg X, \neg Z, \neg D, n=0, m=0\} \mapsto \{\neg E, X, Z\}\,, \tag{27}$$
$$\text{Place-}x\text{-on-}y: \quad \{\neg E, X, Z, \neg D, m=0\} \mapsto \{E, \neg X, \neg Z, A, D\}\,. \tag{28}$$

The system of equations is:

$$1 \;=\; w_E + w_n\,, \tag{29}$$

$$
\begin{aligned}
1 &= w_E + w_m \,, && (30)\\
1 &= w_E - w_Y + w_B \,, && (31)\\
1 &= -w_E \,, && (32)\\
1 &= -w_E + w_Y \,, && (33)\\
1 &= w_E - w_X + w_A \,, && (34)\\
1 &= -w_E + w_X \,, && (35)\\
1 &= w_E - w_X - w_Z \,, && (36)\\
1 &= -w_E + w_X + w_Z - w_A - w_D && (37)
\end{aligned}
$$

The solution is $\{w_E = -1, w_X = 0, w_Y = 0, w_A = 2, w_B = 2, w_D = -4, w_Z = -2, w_n = 2, w_m = 2\}$. Therefore,

$$V^*(s) = 2n(s) + 2m(s) - E(s) + 2A(s) + 2B(s) - 4D(s) - 2Z(s) + 3 \quad (38)$$

since the constant is 3 because at goal: $\{E = 1, X = 0, Y = 0, A = 1, B = 0, D = 1, Z = 0, n = 0, m = 0\}$. For example, starting at a state s where x and y are in the same tower, there are 3 blocks above x but below y, 5 blocks above y, and the gripper is empty, the policy needs

$$
\begin{aligned}
V^*(s) &= 2n(s) + 2m(s) - E(s) + 2A(s) + 2B(s) - 4D(s) - 2Z(s) + 3 && (39)\\
&= 2 \times 3 + 2 \times 5 - 1 + 2 \times 0 + 2 \times 1 - 4 \times 0 - 2 \times 0 + 3 && (40)\\
&= 6 + 10 - 1 + 2 + 3 && (41)\\
&= 20 && (42)
\end{aligned}
$$

steps to reach the goal: 10 steps to clear y, 2 to put y away, 6 to clear x, and 2 to put x on y.

1.1.3 Delivery: One Package

First formulation given in terms of the features $\{H, p, t, n\}$ for holding a package (H), distance to package if not holding package else 0 (p), distance to target cell if holding package else distance from package's cell to target cell (t), and number of packages to deliver (n). The policy is given by the rules $\{\neg H, p > 0\} \mapsto \{p\downarrow\}$, $\{\neg H, p = 0\} \mapsto \{H\}$, $\{H, t > 0\} \mapsto \{t\downarrow\}$, and $\{H, t = 0\} \mapsto \{\neg H, n\downarrow\}$, that induces the system of equations:

$$
\begin{aligned}
1 &= w_p \,, && (43)\\
1 &= -w_H \,, && (44)\\
1 &= w_t \,, && (45)\\
1 &= w_n + w_H && (46)
\end{aligned}
$$

with solution $\{w_p = 1, w_t = 1, w_H = -1, w_n = 2\}$. At the goal all features are zero. The decomposition is

$$V^*(s) = p(s) + t(s) + 2n(s) - H(s) \,. \quad (47)$$

Second formulation given by same features but where t measures the distance to the *target once the package is picked*. The policy rules are $\{\neg H, p > 0\} \mapsto \{p\downarrow\}$, $\{\neg H, p = 0\} \mapsto \{H, t\uparrow\}$, $\{H, t > 0\} \mapsto \{t\downarrow\}$, and $\{H, t = 0\} \mapsto \{\neg H, n\downarrow\}$, where the increment of t is by the amount d_t that measures the distance from the package to the target cell. The equations are:

$$
\begin{aligned}
1 &= w_p \,, && (48)\\
1 &= -d_t w_t - w_H \,, && (49)\\
1 &= w_t \,, && (50)\\
1 &= w_n + w_H \,. && (51)
\end{aligned}
$$

The solution is $\{w_p = 1, w_t = 1, w_H = -(1 + d_t), w_n = 2 + d_t\}$. The decomposition is

$$V^*(s) = (2 + d_t)n(s) + p(s) + t(s) - (1 + d_t)H(s) \,. \quad (52)$$

For example, for the initial state where $\{n = 1, p = d_p, t = 0, H = 0\}$, $V^*(s) = 2 + d_t + d_p$ since the agent must traverse a total distance of $d_t + d_p$ and perform two actions with the package: pick it up and deliver.

1.1.4 Delivery: Many Packages

An instance with multiple packages can be captured with rules $\{\neg H, p>0\} \mapsto \{p\downarrow\}$, $\{\neg H, p=0\} \mapsto \{H, t\uparrow\}$, $\{H, t>0\} \mapsto \{t\downarrow\}$, $\{H, n>0\} \mapsto \{\neg H, n\downarrow, p\uparrow\}$, and $\{H, n>0\} \mapsto \{\neg H, n\downarrow\}$, where the last rule is used only at the end when there are no more packages left and thus p doesn't increment. Under the assumption that all increments of t and p are always by the same amounts d_t and d_p respectively (i.e., all packages are at the same distance from the target cell), the system of equations is:

$$\begin{aligned} 1 &= w_p\,, && (53)\\ 1 &= -d_t w_t - w_H\,, && (54)\\ 1 &= w_t\,, && (55)\\ 1 &= w_n - d_p w_p + w_H\,, && (56)\\ 1 &= w_n + w_H\,. && (57) \end{aligned}$$

The last two equations imply $w_p = 0$ which is inconsistent with the first equation. So, V^* cannot be decomposed using such features. With a slight reformulation of p so that it measures the distance to the next package while there are such packages still left, we can remove the last policy rule and equation. The system then becomes solvable with the solution $\{w_p = 1, w_t = 1, w_H = -(1+d_t), w_n = 2 + d_p + d_t\}$, and the decomposition is

$$V^*(s) \;=\; (2 + d_p + d_t)n(s) + p(s) + t(s) - (1 + d_t)H(s)\,. \qquad (58)$$

1.1.5 Gripper

We consider the policy over the features $\{R, n, g, m\}$ where R tells whether the agent is at room A, n counts the number of balls being held, g counts the number of free grippers, and m counts the number of balls still at room B. The policy is given by the rules $\{R, n=0\} \mapsto \{\neg R\}$, $\{\neg R, g=0\} \mapsto \{R\}$ and $\{\neg R, m=0\} \mapsto \{R\}$ to move between rooms while guaranteeing termination and optimality, $\{R, n>0\} \mapsto \{n\downarrow, g\uparrow\}$ to drop balls being held at room A, and $\{\neg R, m>0, g>0\} \mapsto \{m\downarrow, g\downarrow, n\uparrow\}$ to pick balls at room B. All the increments are by 1. The system of equations is

$$\begin{aligned} 1 &= w_R\,, && (59)\\ 1 &= -w_R\,, && (60)\\ 1 &= -w_R\,, && (61)\\ 1 &= w_n - w_g\,, && (62)\\ 1 &= w_m + w_g - w_n\,. && (63) \end{aligned}$$

The first two equations are inconsistent, and thus V^* cannot be decomposed with such features.

1.1.6 Moving Balls From One Box into Another

Let us consider a variation of Gripper where the agent must move the balls from box B to box A. An optimal policy is obtained with the features $\{n, g, m\}$ where n and g are as in Gripper, and m counts the number of balls still in box B. The policy rules are $\{n>0\} \mapsto \{n\downarrow, g\uparrow\}$ (drop balls being held at box A), and $\{m>0, g>0\} \mapsto \{m\downarrow, g\downarrow, n\uparrow\}$ (pick balls from box B). The system of equations is

$$\begin{aligned} 1 &= w_n - w_g\,, && (64)\\ 1 &= w_m + w_g - w_n\,. && (65) \end{aligned}$$

This time there are multiple solutions for the system of equations that can be expressed in terms of the free variable w_g as $\{w_m = 2, w_n = 1 + w_g\}$. The optimal value function decomposes as

$$V^*(s) \;=\; 2m(s) + (1 + w_g)n(s) + w_g g(s)\,. \qquad (66)$$

1.2 Special Case that Guarantee the Necessary/Sufficient Condition

Definition 5 (Boolean Graph and Acyclicity)**.** *The **Boolean graph** for policy π over the set of features Φ is a directed graph $G = (V, E)$ whose vertices are the joint valuations for the* Boolean features *in Φ (i.e., the numerical features are not considered), and there is edge $(\sigma, \sigma') \in E$ iff the transition (σ, σ') is compatible with a policy rule in π that does not affect any numerical feature. The policy π is **Boolean acyclic** if its Boolean graph is acyclic.*

Definition 6 (Regular Policies)**.** *The policy π is regular over a set of features Φ if the numerical features in Φ can be ordered as $(x_1, \ldots, x_n)$ such that:*

1. *no policy rule in π increments a numerical feature,*
2. *for each numerical feature x_i, there is exactly one policy rule in π that decrements x_i and whose precondition includes $x_j = 0$ for $j = 1, \ldots, i-1$, and*
3. *all numerical features are zero at the goal state.*

For regular policies, we denote the set of features as $\Phi = \{x_1, \ldots, x_n, p_1, \ldots, p_m\}$ where $(x_1, \ldots, x_n)$ is the ordering that satisfy above conditions, and $\{p_1, \ldots, p_m\}$ is the set of Boolean features in Φ.

Lemma 7. *Let π be a regular policy over the features $\Phi = \{x_1, \ldots, x_n, p_1, \ldots, p_m\}$. Then, for each numerical feature x_i, the policy graph for π contains at least one cycle in which the feature x_i is decremented. Furthermore, any cycle that decrements x_i cannot decrement a different numerical feature.*

Proof. The policy graph must have at least one cycle that decrements x_i as otherwise π would not be a solution. Let C be such a cycle, and suppose that another feature x_j is decremented in C. Throughout C, $x_i > 0$ holds and thus, by regularity, $i \leq j$. Swapping the roles of i and j, we obtain $j \leq i$. □

We now establish sufficient conditions for linear decomposition of V^* while given a meaningful interpretation of the weights in the decomposition. For this, we need to "reformulate" a generalized policy π using the possible joint valuations of Boolean features as a brand new set of features.

Definition 8 (Reformulation)**.** *Let π be a regular policy over the set of features $\Phi = \{x_1, \ldots, x_n, p_1, \ldots, p_m\}$, and let $\Phi' = \{x_1, \ldots, x_n, q_1, \ldots, q_{2^m}\}$ be a new set of features where each q_i corresponds to a joint valuation for the m Boolean features in Φ. The policy π over Φ is reformulated into the policy π' over Φ' as follows. For each edge (σ, σ') in the policy graph for π that is compatible with the policy rule a, where $\sigma \vDash q$ and $\sigma' \vDash q'$, π' contains the policy rule $\{q\} \mapsto \{\neg q, q'\}$ or $\{q\} \mapsto \{\neg q, q', x_i\downarrow\}$ whether a decrements x_i or not.*

The set of equations for the reformulated policy π' consists of two type of equations:

$$1 = w_q - w_{q'}\,, \tag{67}$$

$$1 = w_{x_i} + w_q - w_{q'}\,. \tag{68}$$

The first type of equations (67) correspond to policy rules of the form $\{q\} \mapsto \{\neg q, q'\}$, while the second type of equations (68) correspond to rules of the form $\{q\} \mapsto \{\neg q, q', x_i\downarrow\}$. The policy graph for π' contains vertices for the joint valuations of the q_i features. However, any state in the problem makes true exactly one such feature, and vertices in the policy graph that make true two different q features can be safely removed. The **reduced policy graph** thus contains at most 2^m different vertices for each joint Boolean valuation of the numerical features, and it is isomorphic to the policy graph for π.

Definition 9 (Uniform Cycles and Sinks)**.** *A policy π has **uniform cycles** if for any numerical feature x, and any two cycles C and C' in the policy graph where x is decremented, the length of both cycles is the same. A Boolean graph has **uniform sinks** if for any vertex q, and any two sink vertices q' and q'' that are reachable from q, the distances from q to q' and q'' are the same. A policy pi is **uniform** if it has uniform cycles and its Boolean graph has uniform sinks.*

Theorem 10. *Let π be a regular and deterministic policy, and let π' be the reformulated policy. If π' is uniform, the optimal value function V^* decomposes linearly over Φ'. Furthermore, there is a decomposition where the weight w_q is equal to the distance from q to the unique sink reachable from q in the Boolean graph G, and the weight w_x is equal to the length of any cycle that decrements x.*

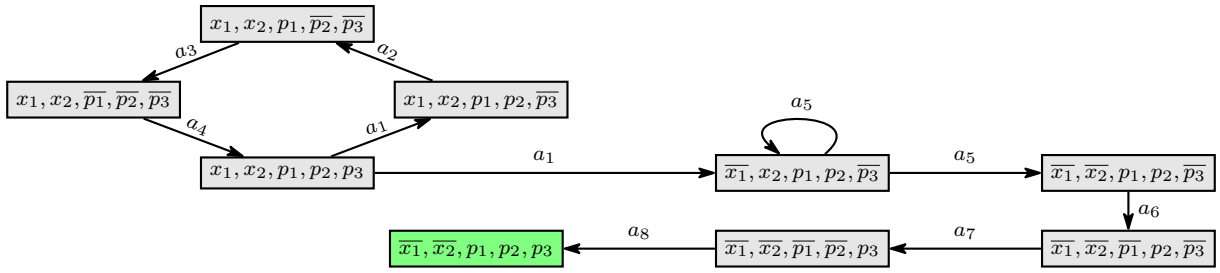

Figure 1: Policy graph for the policy π over the features $\Phi = \{x_1, x_2, p_1, p_2, p_3\}$. Literal $\neg p$ is denoted by $\overline{p}$, and literals $x > 0$ and $x = 0$ are denoted by x and $\overline{x}$. Policy rules are: $a_1 : \{p_1, p_2, p_3, x_1\} \mapsto \{\overline{p_3}, x_1\downarrow\}$, $a_2 : \{p_1, p_2, \overline{p_3}, x_1\} \mapsto \{\overline{p_2}\}$, $a_3 : \{p_1, \overline{p_2}, \overline{p_3}\} \mapsto \{\overline{p_1}\}$, $a_4 : \{\overline{p_1}, \overline{p_2}, \overline{p_3}\} \mapsto \{p_1, p_2, p_3\}$, $a_5 : \{p_1, p_2, \overline{p_3}, \overline{x_1}, x_2\} \mapsto \{x_2\downarrow\}$, $a_6 : \{p_1, p_2, \overline{p_3}, \overline{x_1}, \overline{x_2}\} \mapsto \{\overline{p_1}\}$, $a_7 : \{\overline{p_1}, p_2, \overline{p_3}, \overline{x_1}, \overline{x_2}\} \mapsto \{\overline{p_2}\}$, $a_8 : \{\overline{p_1}, \overline{p_2}, p_3, \overline{x_1}, \overline{x_2}\} \mapsto \{p_1, p_2\}$.

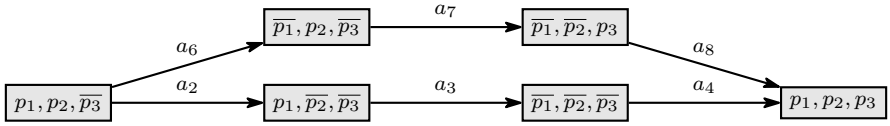

Figure 2: Policy graph for the policy π over the features $\Phi = \{x_1, x_2, p_1, p_2, p_3\}$. Literal $\neg p$ is denoted by $\overline{p}$, and literals $x > 0$ and $x = 0$ are denoted by x and $\overline{x}$.

Proof. Let us partition the policy rules in π' into those that affect numerical features and those that only affect Boolean features. The system of equations for the latter set is called the *Boolean part*. By regularity of π, the system of linear equations is solvable iff the Boolean part is solvable. We show that the Boolean part is solvable if the policy graph is Boolean acyclic.

We construct a solution for the Boolean part inductively using the Boolean graph G. For the sink vertices q in G, define $w_q := 0$, while for non-sink vertices q, after the unique successor q' of q in G has received a value, define $w_q := 1 + w_{q'}$. Clearly, such weights are well defined (since G is acyclic and has uniform cycles), and provide a solution to the Boolean part.

It is not difficult to show that the weight w_q measures the distance in the Boolean graph G from q to any sink vertex q' that is reachable from q. On the other hand, a policy rule of form $\{q, x_i > 0\} \mapsto \{\neg q, q', x_i\downarrow\}$, associated with the equation $1 = w_{x_i} + w_q - w_{q'}$, corresponds to a pair of *non-deterministic edges* in the policy graph. In such a case, the weight for x_i is set to $w_{x_i} = 1 - w_q + w_{q'}$ according to the equation for the policy rule. We now show that G contains a path from q' to q.

Let C be a cycle in the policy graph where the rule a is applied; that is, $C = (\sigma, \sigma', \ldots, \sigma)$ where $\sigma \vDash q$, $\sigma' \vDash \sigma'$, the transition (σ, σ') is compatible with a, and all the numerical features have the same Boolean valuation throughout C. (**** NEED EXTRA CONDITION: x cannot be decremented twice in C ****) By Lemma 2 and the uniform cycles in π, the path $\sigma \rightsquigarrow \sigma'$ does not contain any edge associated with a rule that decrements any feature. Thus, the Boolean graph G must contain a path from q' to q. Notice that the vertex q in G must be a sink as no other policy rule can be applied at q by determinism. Hence, $w_q = 0$, w'_q measures the distance from q' to q in G, and $w_{x_i} = |C|$. Finally, by uniformity, $|C|$ equals the length of any cycle in which x_i is decremented. □

Let see an example of Theorem 10. Figure 1 shows policy graph for the policy described in the caption, Fig. 2 shows the Boolean graph, which is acyclic, and Fig. 3 shows the reduced policy graph for the reformulated policy. Since all features have value of zero at the goal state (green state), the constant for the linear decomposition is zero. By Theorem 10, the weigths are $\{w_{x_1} = 4, w_{x_2} = 1, w_{q_0} = 1, w_{q_1} = 1, w_{q_2} = 2, w_{q_4} = 2, w_{q_6} = 3, w_{q_7} = 0\}$, and the decomposition of V^* becomes

$$V^*(s) \;=\; 4x_1(s) + x_2(s) + q_0(s) + q_1(s) + 2q_2(s) + 2q_4(s) + 3q_6(s)\,. \tag{69}$$

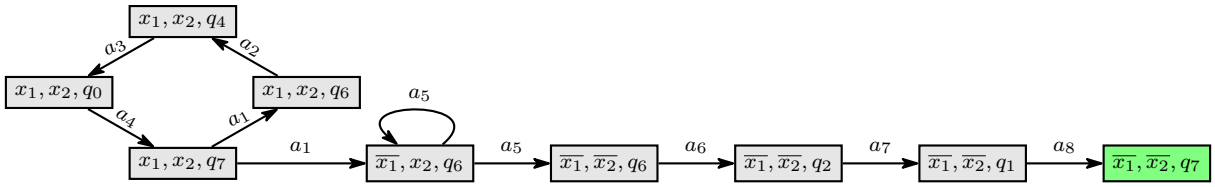

Figure 3: Reduced policy graph for the reformulated policy π' over $\Phi = \{x_1, x_2, q_0, q_1, q_2, q_4, q_6, q_7\}$. Literal $\neg p$ is denoted by $\overline{p}$, and literals $x > 0$ and $x = 0$ are denoted by x and $\overline{x}$. Policy rules are: $a_1 : \{q_7, x_1\} \mapsto \{\overline{q_7}, q_6, x_1\downarrow\}$, $a_2 : \{q_6, x_1\} \mapsto \{\overline{q_6}, q_4\}$, $a_3 : \{q_4\} \mapsto \{\overline{q_4}, q_0\}$, $a_4 : \{q_0\} \mapsto \{\overline{q_0}, q_7\}$, $a_5 : \{q_6, \overline{x_1}, x_2\} \mapsto \{x_2\downarrow\}$, $a_6 : \{q_6, \overline{x_1}, \overline{x_2}\} \mapsto \{\overline{q_6}, q_2\}$, $a_7 : \{q_2, \overline{x_1}, \overline{x_2}\} \mapsto \{\overline{q_2}, q_1\}$, $a_8 : \{q_1, \overline{x_1}, \overline{x_2}\} \mapsto \{\overline{q_1}, q_7\}$.

1.2.1 General case

Vertices in the policy graph are denoted by σ. We consider features $\langle\sigma\rangle$ that at state s evaluates to 1 if $s \vDash \sigma$ else 0, and features $\langle\sigma\rangle x$ that at state s evaluates to $x(s)$ if $s \vDash \sigma$ else 0. Observe that $\langle\sigma\rangle x(s) = \langle\sigma\rangle(s)\, x(s)$.

Definition 11 (Regular Policy Graph)**.** *A policy graph over features in Φ is **regular** if the following conditions are met:*

1. *Each σ appears in at most one cycle. The cycle where σ appears is denoted by C_σ (i.e. a sequence of σ's). If no cycle contains σ, C_σ denotes the empty sequence. The length of a cycle C is denoted by $|C|$, and the length of the empty sequence is 0.*
2. *Each cycle C has **exactly one** "exit point", where an exit point of C is some σ in C such that the policy rule for σ decreases some numerical feature x. The exit point of the cycle C_σ is denoted by $exit_\sigma$. If C_σ is empty, $exit_\sigma$ is defined as σ.*
3. *For any transition (σ, σ'), there is at most one numerical feature x on which σ and σ' differ. This implies that if σ is an exit point of a cycle C, the policy rule for σ decreases **exactly one** numerical feature. The numerical feature decreased in the exit point for the cycle C_σ is denoted by x_σ. If C_σ is empty, x_σ denotes any (fixed) numerical feature in Φ.*
4. *If the numerical feature x is decremented in the cycle C, there is no σ reachable from but outside C where x is decremented.*
5. *If the policy decrements a variable in σ, σ belongs to some cycle.*
6. *No goal σ appears inside a cycle.*

The policy π is regular is its policy graph is regular.

Theorem 12. *If π is a **regular** policy over the features Φ, V_π decomposes linearly on the set of features $\{\langle\sigma\rangle x : \sigma, x\} \cup \{\langle\sigma\rangle : \sigma\}$. The decomposition can be expressed as $V_\pi = \sum_\sigma \langle\sigma\rangle f_\sigma$ where the functions f_σ can be computed by **backward induction** using the policy graph as follows:*

$$f_\sigma \;=\; \begin{cases} 0 & \sigma \text{ is goal,} \\ dist(\sigma, exit_\sigma) + |C_\sigma|(x_\sigma - 1) + 1 + f_{\sigma'} & \text{otherwise.} \end{cases} \tag{70}$$

In this expression, $dist(\sigma, exit_\sigma)$ is the length of the path $\sigma \rightsquigarrow exit_\sigma$ (if $exit_\sigma = \sigma$, the length is 0), and σ' is the unique successor of $exit_\sigma$ that lies outside *the cycle C_σ (if C_σ is empty, σ' is the unique successor of σ).*

For example, let us consider the policy graph in Fig. 3 for the reduced policy π'. The following table contains the different σ in the policy graph, and the corresponding functions f_σ:

σ	Boolean valuation	f_σ
σ_0	$\langle x_1, x_2, q_7\rangle$	$f_{\sigma_0} = 4(x_1 - 1) + 1 + f_{\sigma_4} = 4x_1 + x_2$
σ_1	$\langle x_1, x_2, q_6\rangle$	$f_{\sigma_1} = 3 + 4(x_1 - 1) + 1 + f_{\sigma_4} = 4x_1 + x_2 + 3$
σ_2	$\langle x_1, x_2, q_4\rangle$	$f_{\sigma_2} = 2 + 4(x_1 - 1) + 1 + f_{\sigma_4} = 4x_1 + x_2 + 2$
σ_3	$\langle x_1, x_2, q_0\rangle$	$f_{\sigma_3} = 1 + 4(x_1 - 1) + 1 + f_{\sigma_4} = 4x_1 + x_2 + 1$
σ_4	$\langle \overline{x_1}, x_2, q_6\rangle$	$f_{\sigma_4} = x_2 + f_{\sigma_5} = x_2 + 3$
σ_5	$\langle \overline{x_1}, \overline{x_2}, q_6\rangle$	$f_{\sigma_5} = 3$
σ_6	$\langle \overline{x_1}, \overline{x_2}, q_2\rangle$	$f_{\sigma_6} = 2$
σ_7	$\langle \overline{x_1}, \overline{x_2}, q_1\rangle$	$f_{\sigma_7} = 1$
σ_8	$\langle \overline{x_1}, \overline{x_2}, q_7\rangle$	$f_{\sigma_8} = 0$

The decomposition of V_π provided by Theorem 12 is then:

$$\begin{aligned} V_\pi &= \langle\sigma_0\rangle f_0 + \langle\sigma_1\rangle f_1 + \langle\sigma_2\rangle f_2 + \langle\sigma_3\rangle f_3 + \langle\sigma_4\rangle f_4 + \langle\sigma_5\rangle f_5 + \langle\sigma_6\rangle f_6 + \langle\sigma_7\rangle f_7 + \langle\sigma_8\rangle f_8 && (71)\\ &= 4\big(\langle\sigma_0\rangle + \langle\sigma_1\rangle + \langle\sigma_2\rangle + \langle\sigma_3\rangle\big)x_1 + \big(\langle\sigma_0\rangle + \langle\sigma_1\rangle + \langle\sigma_2\rangle + \langle\sigma_3\rangle + \langle\sigma_4\rangle\big)x_2 + && (72)\\ &\quad 3\big(\langle\sigma_1\rangle + \langle\sigma_4\rangle + \langle\sigma_5\rangle\big) + 2\big(\langle\sigma_2\rangle + \langle\sigma_6\rangle\big) + \big(\langle\sigma_3\rangle + \langle\sigma_7\rangle\big) && (73)\\ &= 4x_1 + x_2 + 3q_6 + 2q_4 + 2q_2 + q_1 + q_0\,. && (74)\end{aligned}$$

This decomposition is exactly the same as in (69).

Proof of Theorem 12. First observe that for any σ, $f_\sigma(s) = f_\sigma(s')$ for any pair of states s and s' that agree on the values of the numerical features in Φ. Likewise, if σ belongs to cycle C_σ and σ' is reachable from σ but outside C_σ, $f_{\sigma'}(s') = f_{\sigma'}(s)$ if s and s' agree on the values of the numerical features except x_σ (this follows by the condition 4 for regular policies).

We show by backward induction $V_\pi(s) = \sum_\sigma(\langle\sigma\rangle f_\sigma)(s)$. The base case is for goal states s. In this case, $\sigma(s) = 1$ only if σ is a goal vertex in the policy graph and $f_\sigma = 0$ by definition. Clearly, $V_\pi(s) = \sum_\sigma(\langle\sigma\rangle f_\sigma)(s) = 0$. Let us assume that the claim holds for any state s' with $V_\pi(s') < d$, and let s be a state such that $V_\pi(s) = d$. If s does not belong to any cycle, it does not decrement any feature and thus it has a unique successor s' such that $V_\pi(s') = d - 1$, and s and s' agree on the value of the numerical features. The policy graph contains a unique edge (σ, σ') such that $\sigma \vDash s$ and $\sigma' \vDash s'$. Therefore, using the inductive hypothesis and $f_{\sigma'}(s') = f_{\sigma'}(s)$,

$$V_\pi(s) \;=\; 1 + V_\pi(s') \;=\; 1 + f_{\sigma'}(s') \;=\; 1 + f_{\sigma'}(s) \;=\; f_\sigma(s) \;=\; \sum_\sigma(\langle\sigma\rangle f_\sigma)(s)\,. \qquad (75)$$

The other case is when σ belongs to a cycle, the unique cycle C_σ since π is regular. The execution starting at s exits the cycle when it reaches a state s' such that $s \vDash \text{exit}_\sigma$ and $x_\sigma(s) = 1$. Once at s', the feature x_σ is decremented and the execution continues at state s'' such that $s'' \vDash \sigma'$ for the unique σ' outside C_σ that follows exit_σ. The number of steps for reaching s'' is thus the number of steps to reach exit_σ, plus the number of steps along all iterations of the cycle C_σ, plus 1 more step to reach s''; that is,

$$\begin{aligned} V_\pi(s) &= dist(\sigma, \text{exit}_\sigma) + |C_\sigma|(x_\sigma(s) - 1) + 1 + V_\pi(s'') && (76)\\ &= dist(\sigma, \text{exit}_\sigma) + |C_\sigma|(x_\sigma(s) - 1) + 1 + f_{\sigma'}(s'')\,. && (77)\end{aligned}$$

To complete the proof, by the definition of f_σ, it is enough to show $f_{\sigma'}(s'') = f_{\sigma'}(s)$. Yet this is direct by the observations at the beginning of the proof since σ' is reachable from and outside C_σ, and s and s'' agree on the values for all the numerical features except x_σ. □

Corollary 13. *If π is an **optimal** and **regular** policy, V^* decomposes linearly on the set of features $\{\langle\sigma\rangle x : \sigma, x\} \cup \{\langle\sigma\rangle : \sigma\}$.*

Proof. By Theorem 12, V_π decomposes linearly over the set of features. By optimality, $V^* = V_\pi$. □