دانشگاه صنعتی امیرکبیر
دانشکده مهندسی کامپیوتر

مبانی و کاربردهای هوش مصنوعی

جلسه سوم کلاس حل تمرین

دی ۱۴۰۰

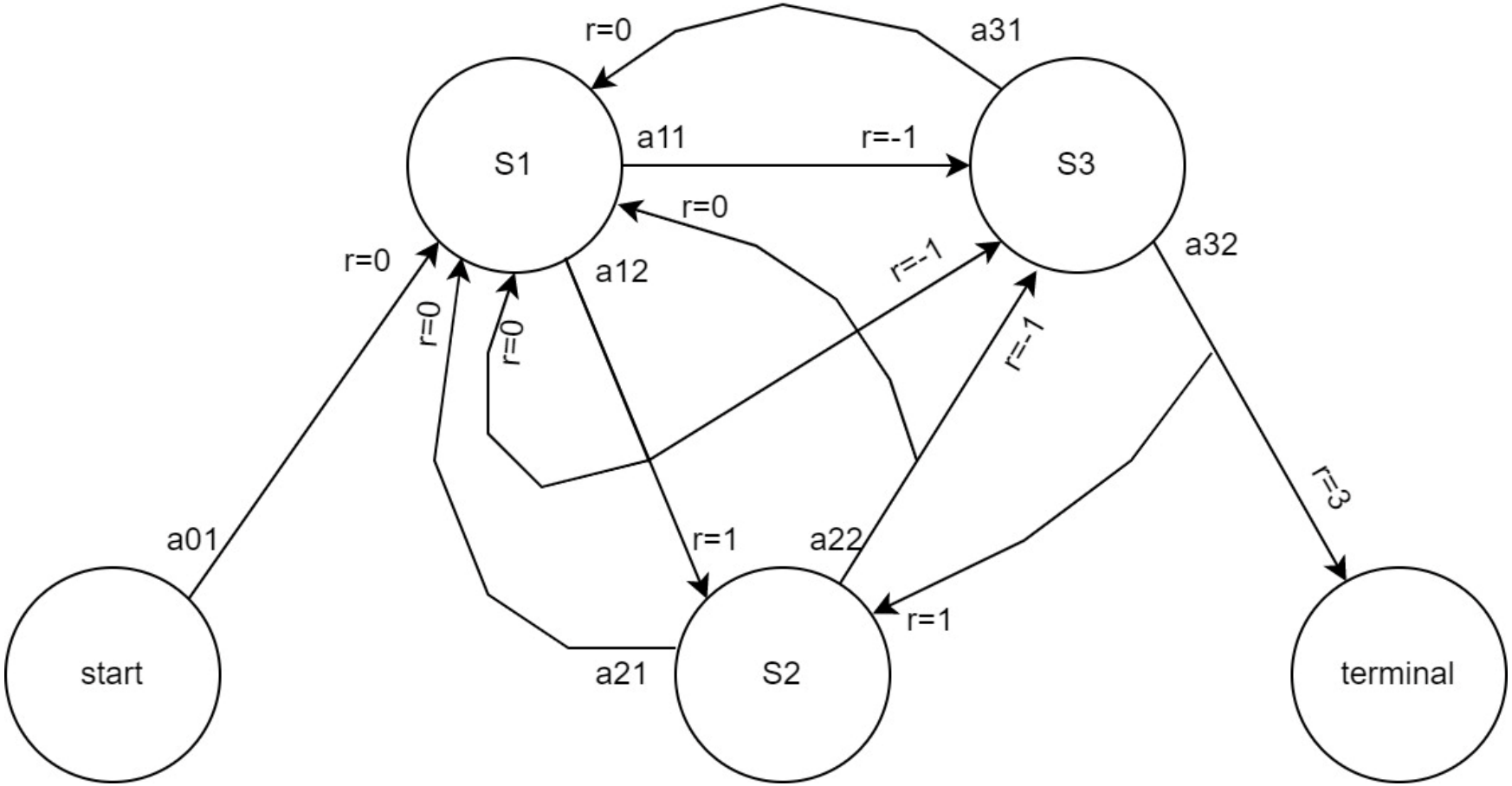
فرایند تصمیم مارکو یا MDP

اجزای سازنده یک MDP شامل:

- مجموعه ای از وضعیت‌ها(s)
- مجموعه ای از اعمال (a)
- یک تابع تبدیل، یا مدلی از احتمال انتقال از یک وضعیت به وضعیت های دیگر ((T(s,a,s’
- یک تابع پاداش ((R(s,a,s’
- وضعیت شروع و (گاها) پایانی

- یک راه برای حل مسائل جستجوی غیر قطعی
- نتایج مورد انتظار برآورد می‌شود و مطابق با آن عمل صورت می‌گیرد

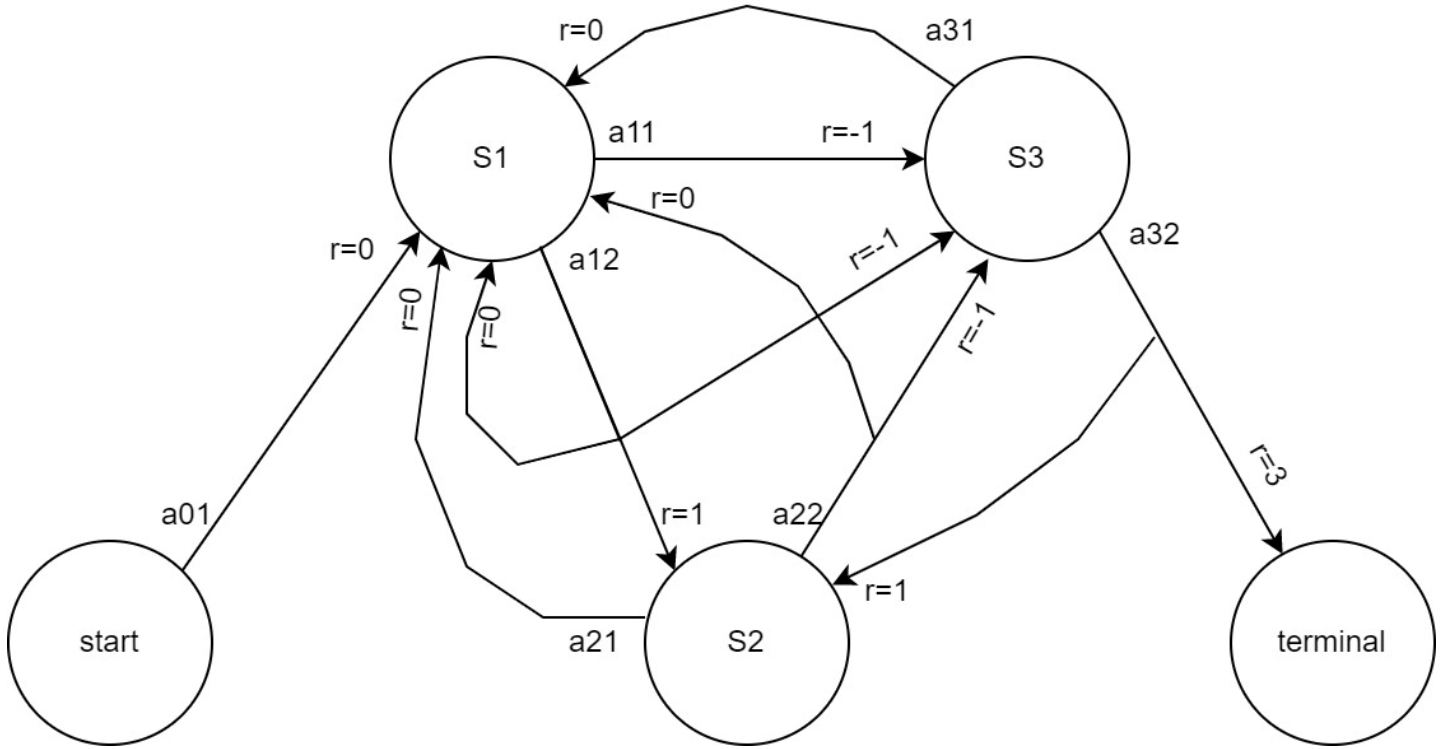
خاصیت مارکو: استقلال از گذشته



سیاست (Policy)

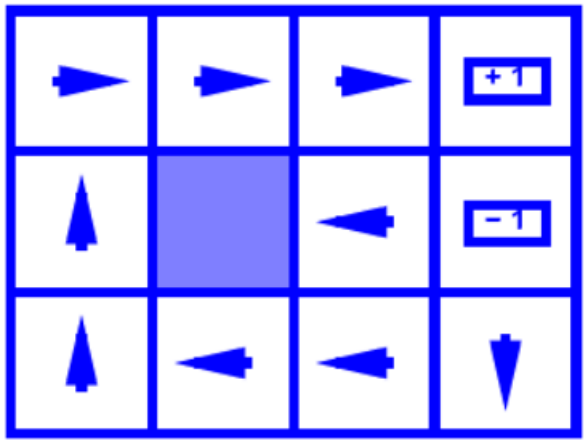
سیاست $\prod$ به ما نقشه راه را نشان می‌دهد

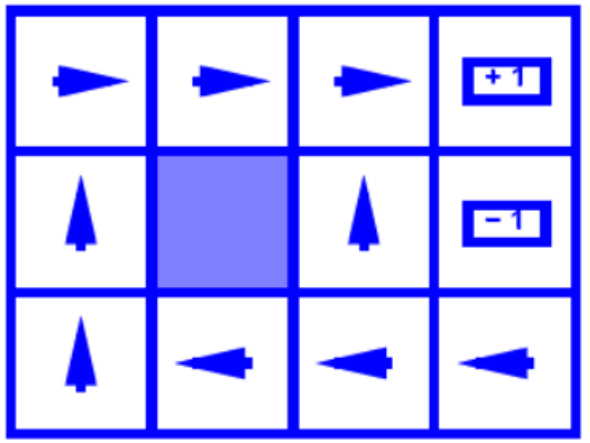
- به ازای هر وضعیت، یک عمل
- هدف نهایی: کشف سیاست بهینه

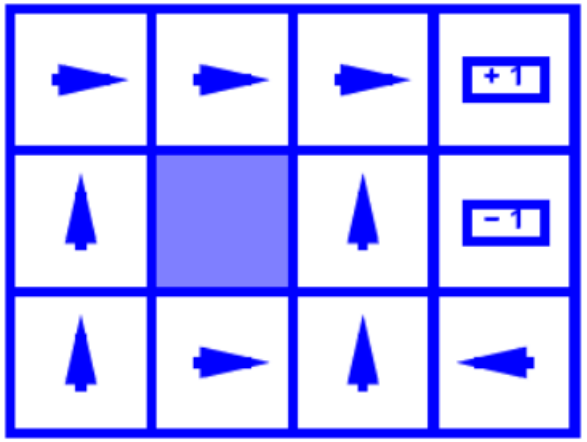


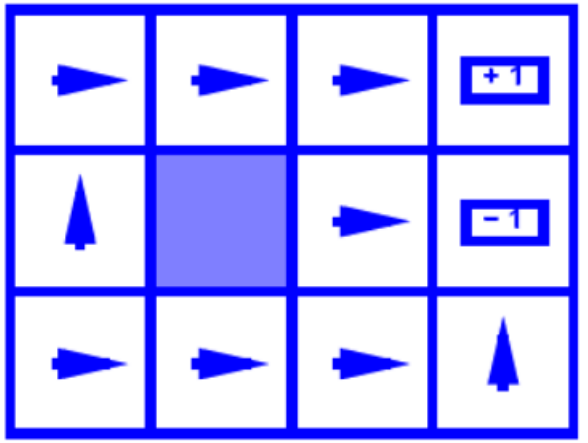
Start: a01
S1: a12
S2: a22
S3: a32

Different Policies

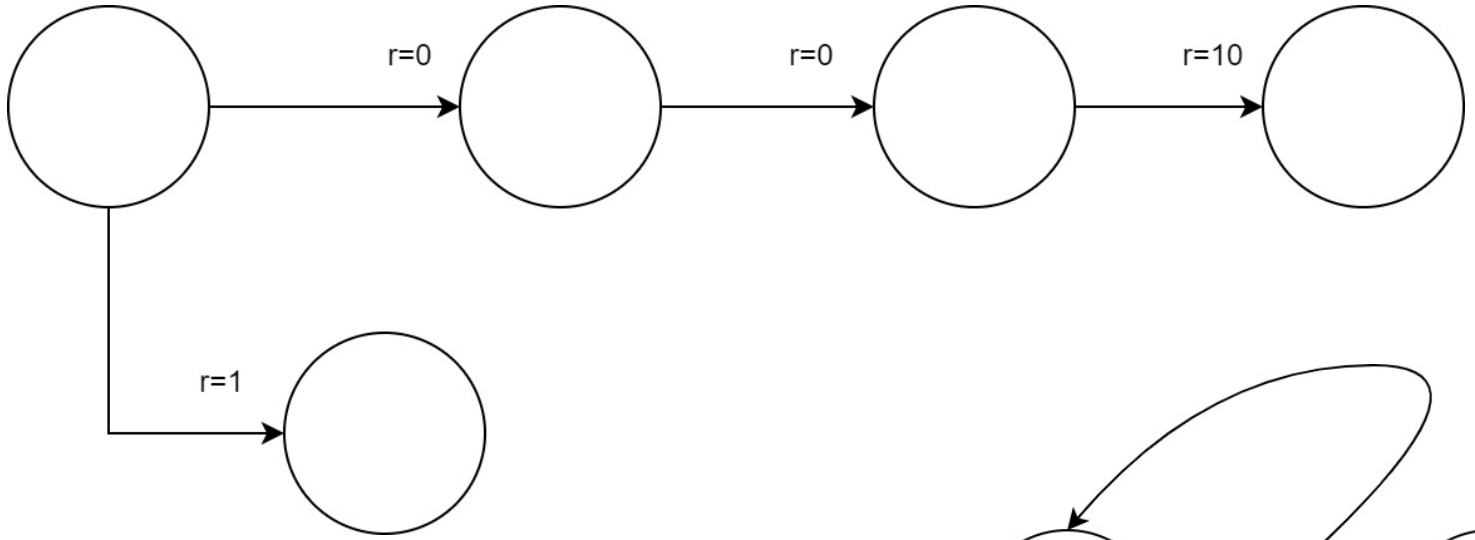


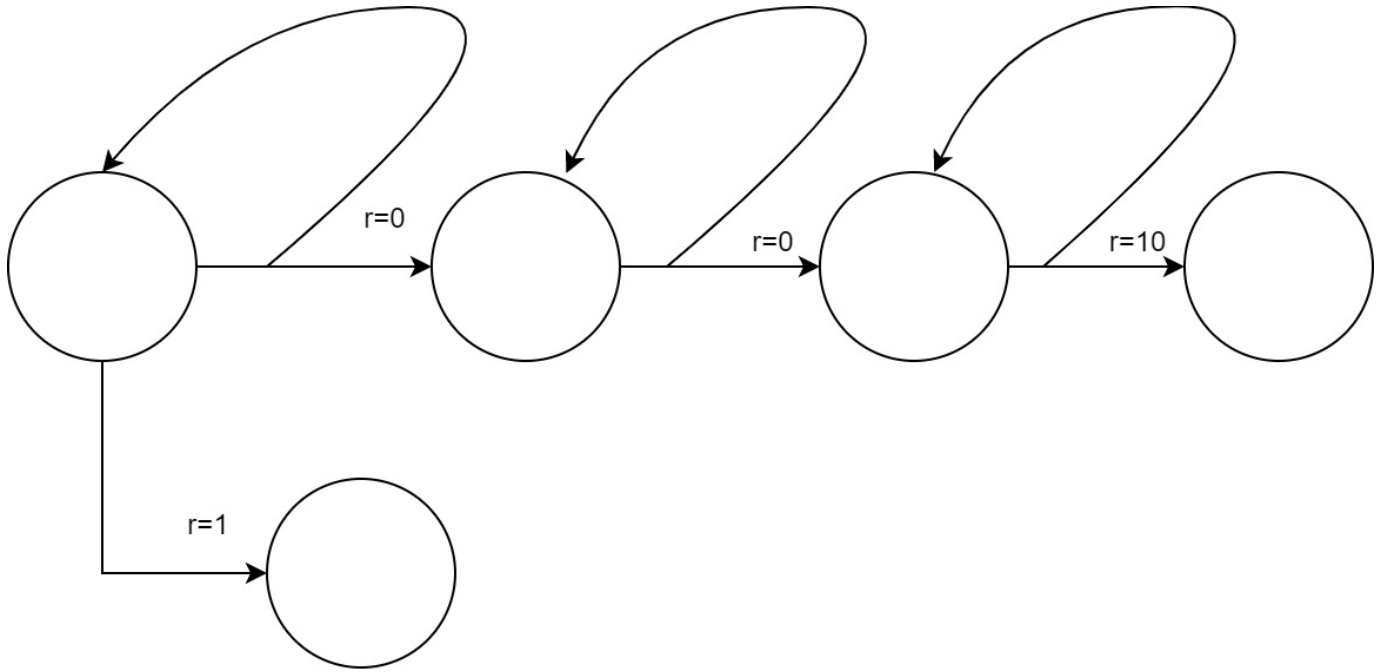






- پاداش زود یا دیر؟





تعیین γ متناسب با نیاز
حل MDP با توجه به احتمالات

حل MDP

مقادیر بهینه:

V*(S) یا مقدار مورد انتظار بهینه در وضعیت S
Q*(S,A) یا مقدار مورد انتظار بهینه به ازای عمل A در وضعیت S
∏*(S) یا عمل بهینه در وضعیت S

$$V^*(s) = \max_a Q^*(s,a)$$

$$Q^*(s,a) = \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V^*(s')\right]$$

$$V^*(s) = \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V^*(s')\right]$$

- به دست آوردن هر کدام از موارد بالا معادل حل شدن MDP است

Value iteration:

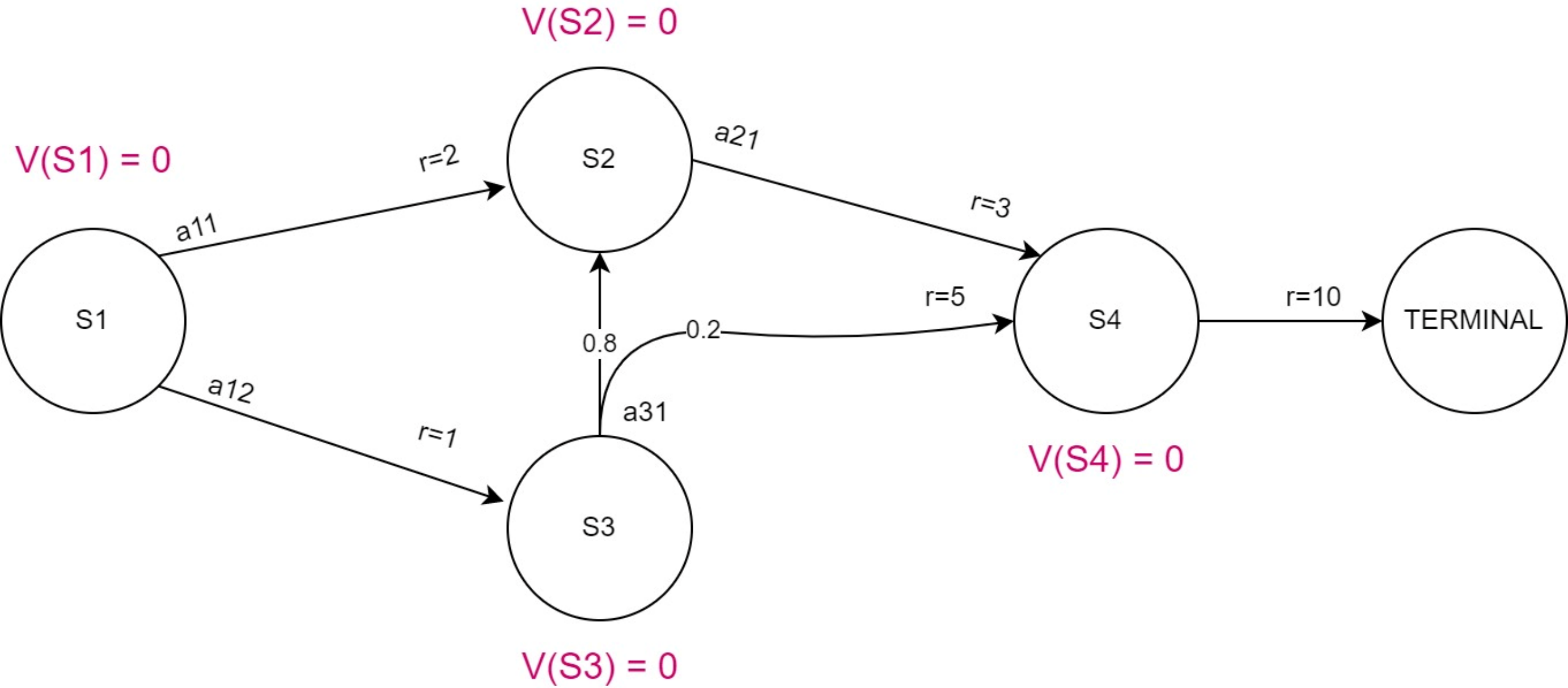
فرض می کنیم همه مقادیر V برای همه وضعیت ها صفر است.

- انتظار خاصی از هیچ وضعیتی نداریم

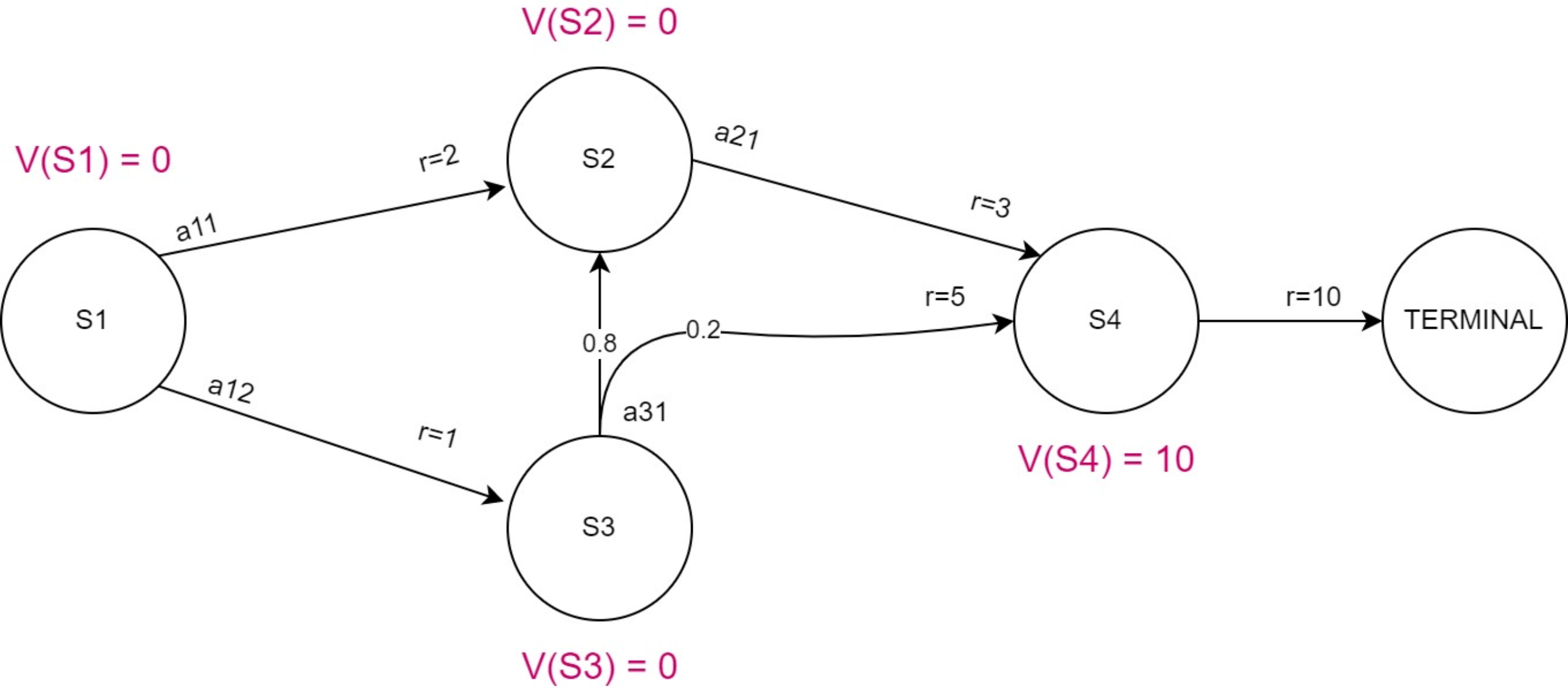
به صورت متناوب همه وضعیت هارا طبق فرمول زیر به روز می کنیم:

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_k(s') \right]$$

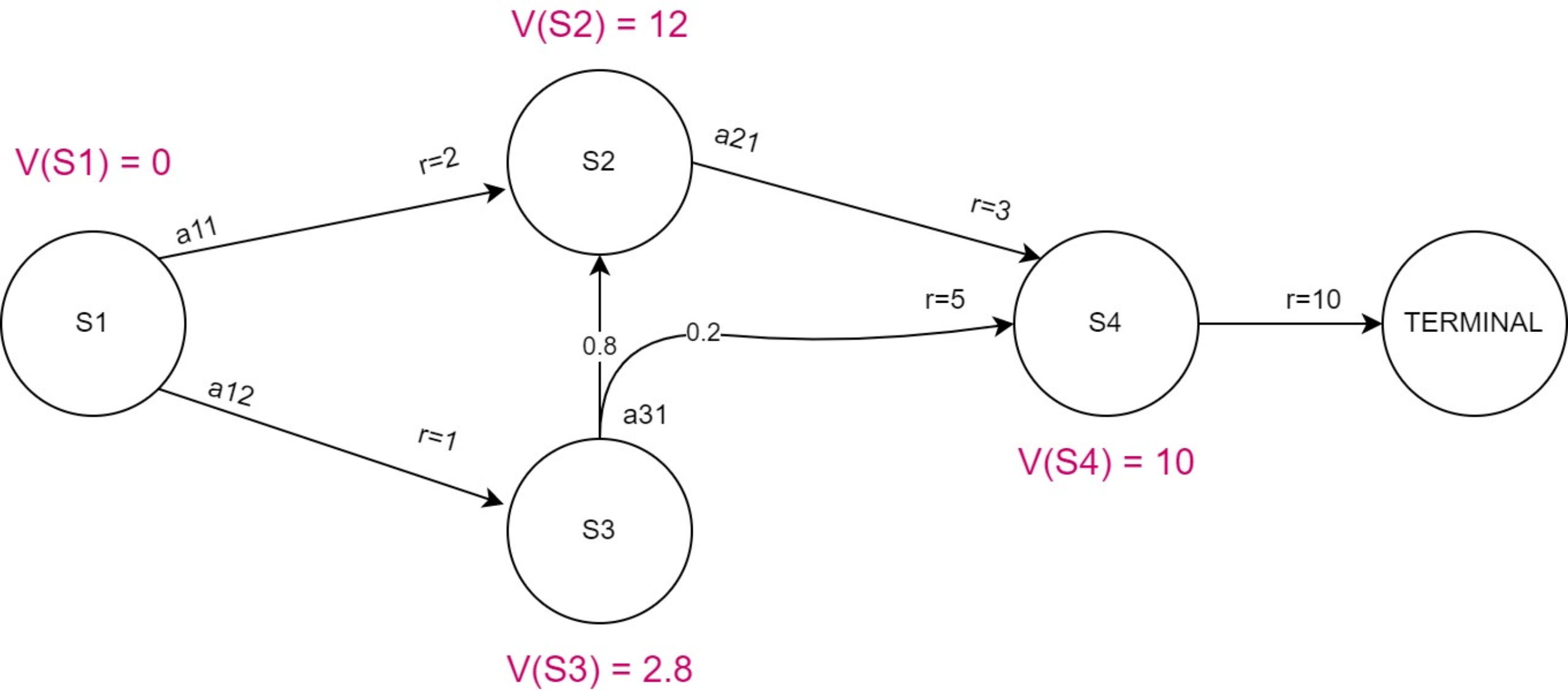
تا زمانی که همگرا شوند ادامه می‌دهیم
اثبات می‌شود که مقادیر به دست آمده تحت MDP مد نظر بهینه خواهند بود



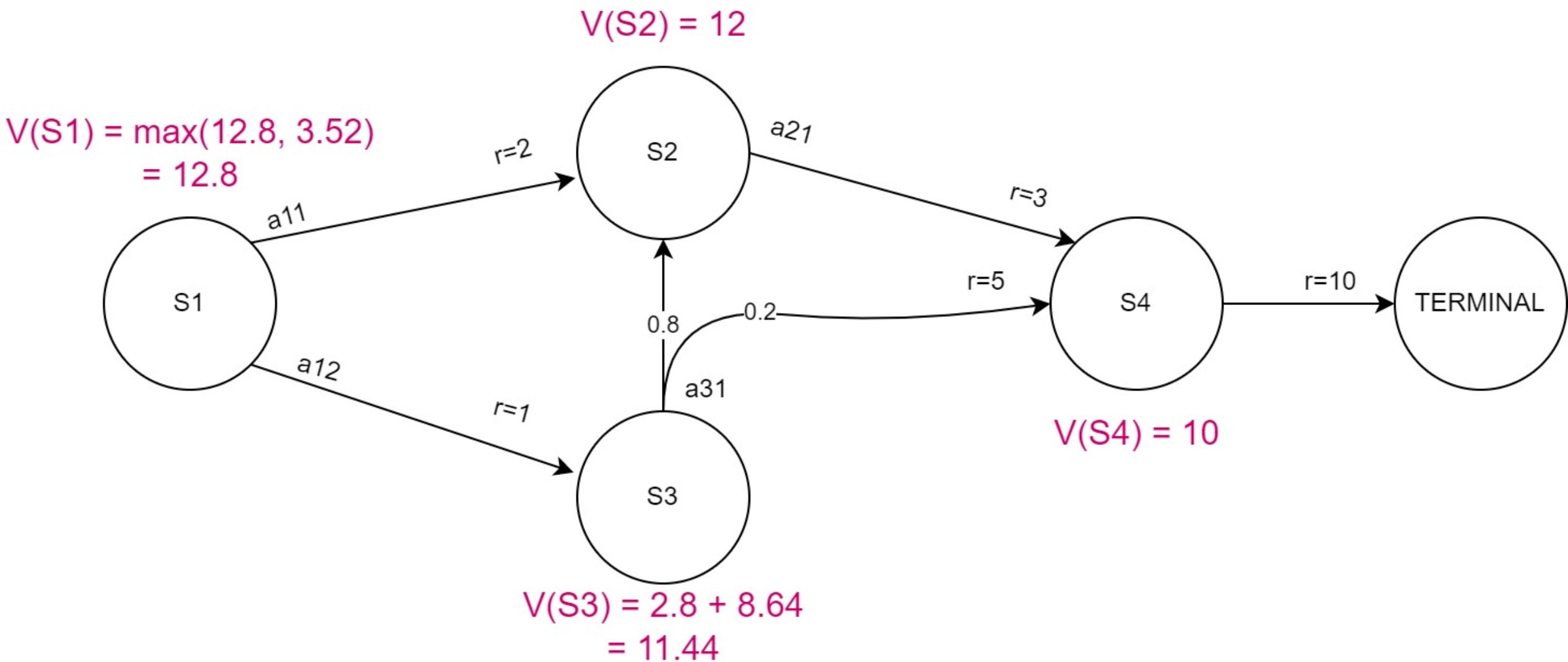
$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_k(s') \right]$$



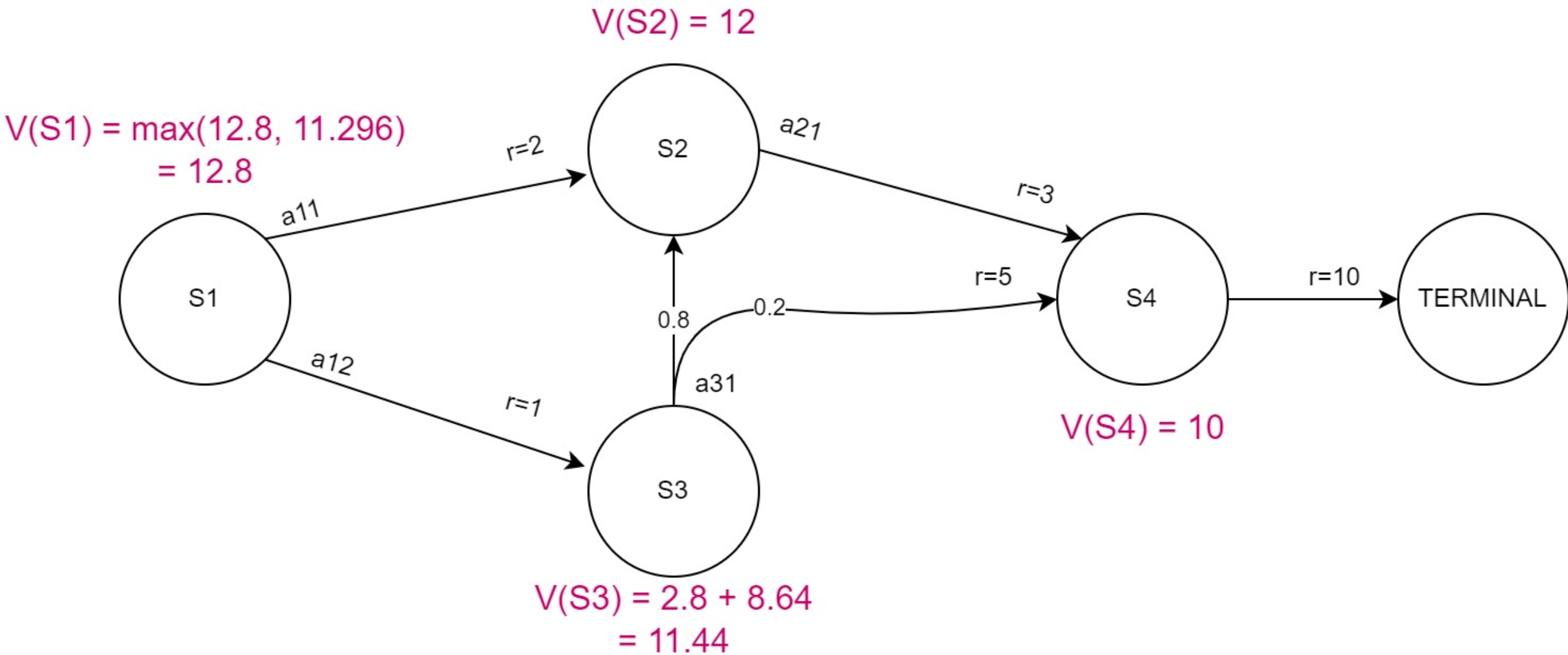
$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_k(s') \right]$$



$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_k(s') \right]$$



$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_k(s') \right]$$



$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_k(s') \right]$$

Policy iteration:

دو قدم زیر را تا زمانی که سیاست همگرا شود (تغییری نکند) ادامه می‌دهیم

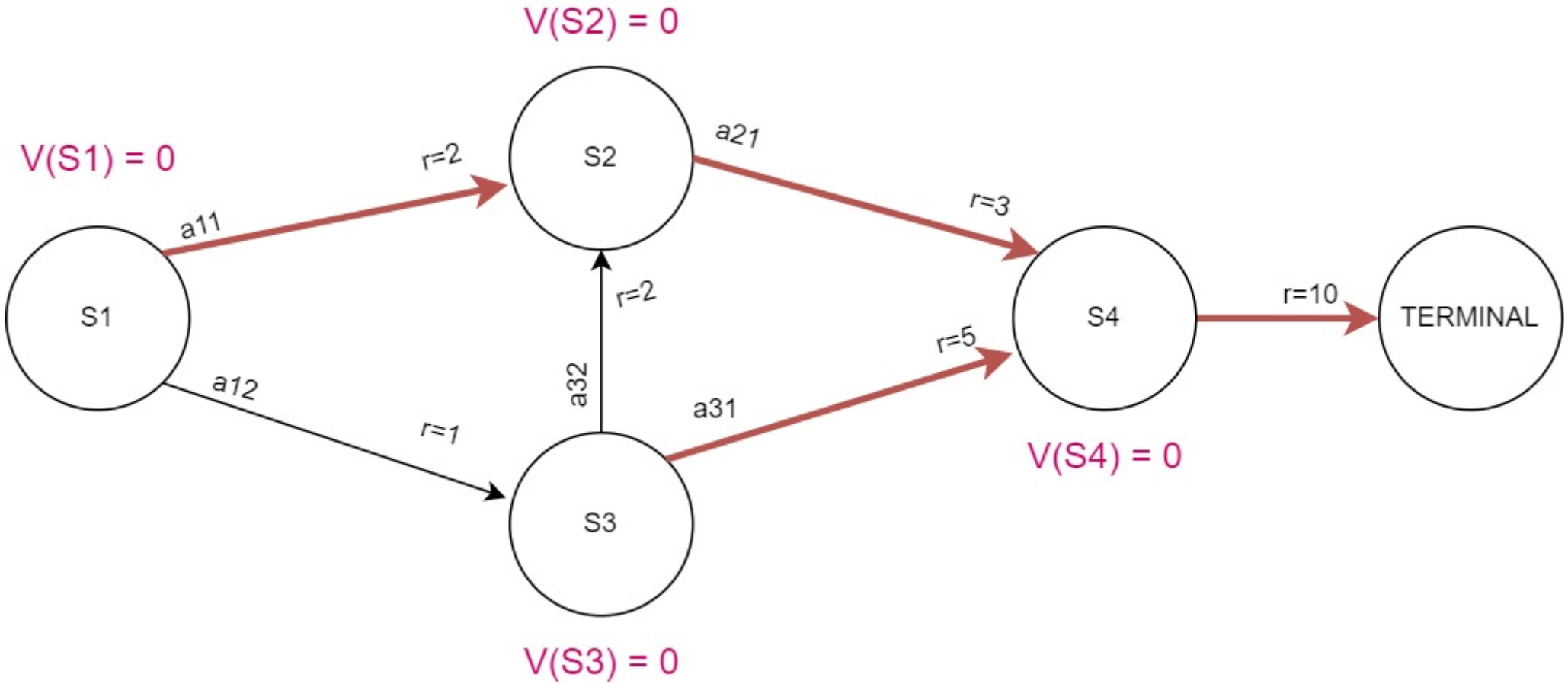
- مقادیر V را برای سیاست فعلی (نه الزاما بهینه) محاسبه می کنیم.

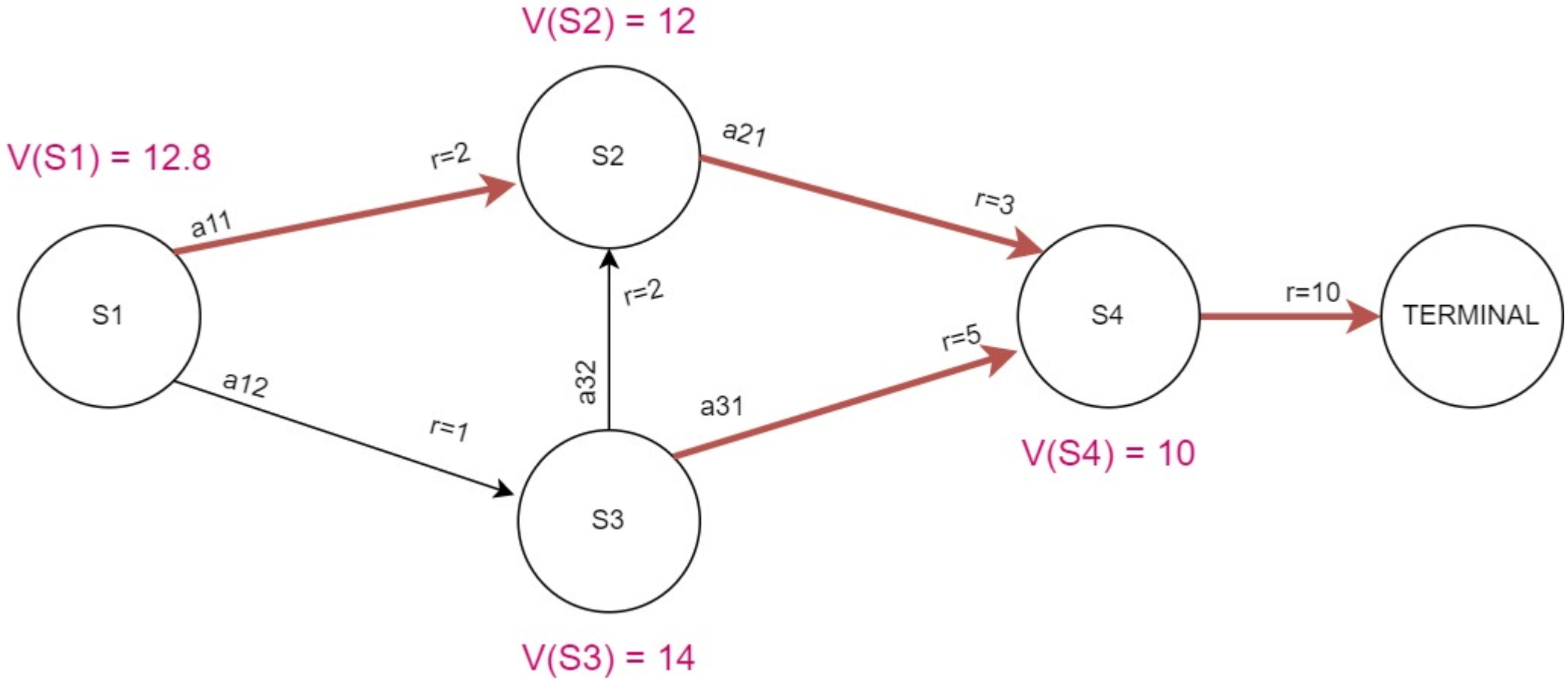
$$V_{k+1}^{\pi_i}(s) \leftarrow \sum_{s'} T(s, \pi_i(s), s') \left[R(s, \pi_i(s), s') + \gamma V_k^{\pi_i}(s') \right]$$

- بر اساس V های به دست آمده، سیاست را به روز می کنیم.

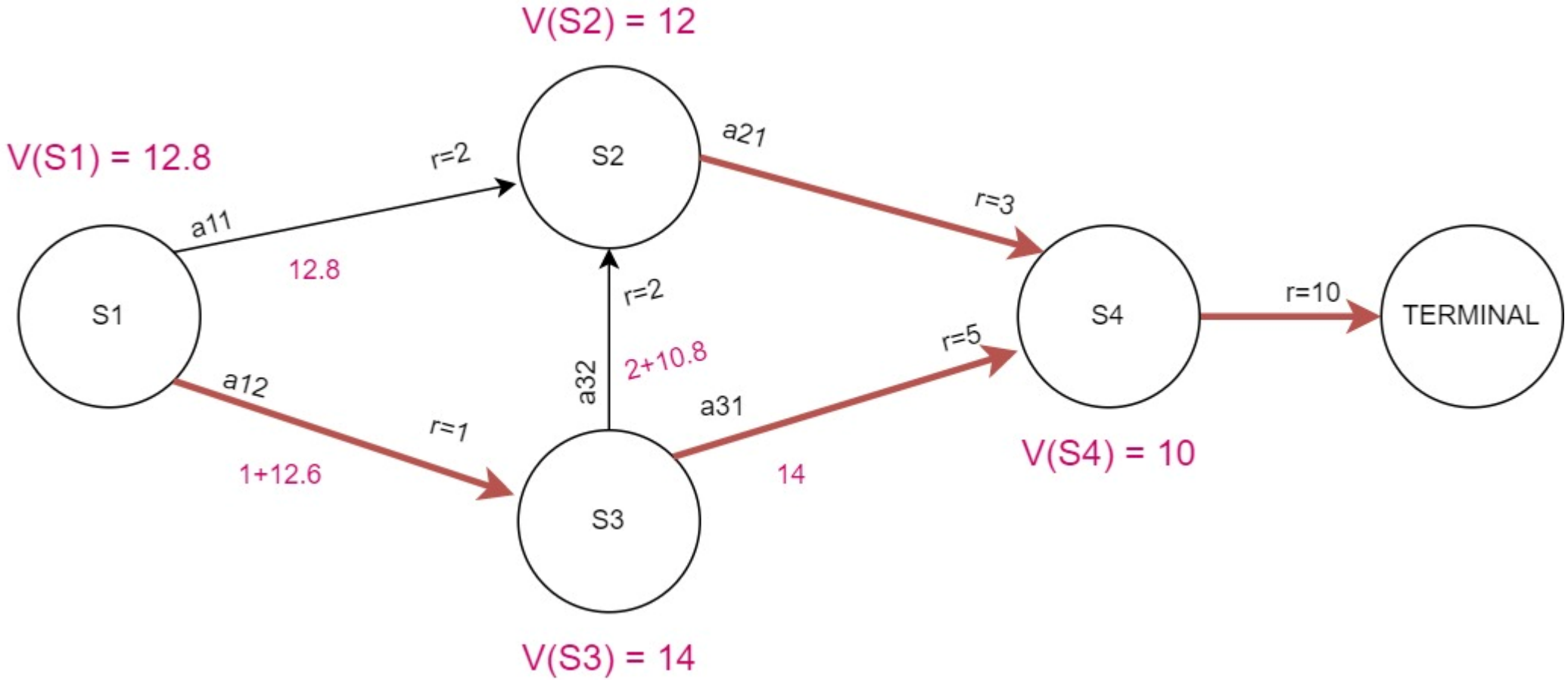
$$\pi_{i+1}(s) = \arg\max_a \sum_{s'} T(s, a, s') \left[R(s, a, s') + \gamma V^{\pi_i}(s') \right]$$

این روش معادل Value iteration است، منتهی در برخی شرایط زودتر همگرا می شود.

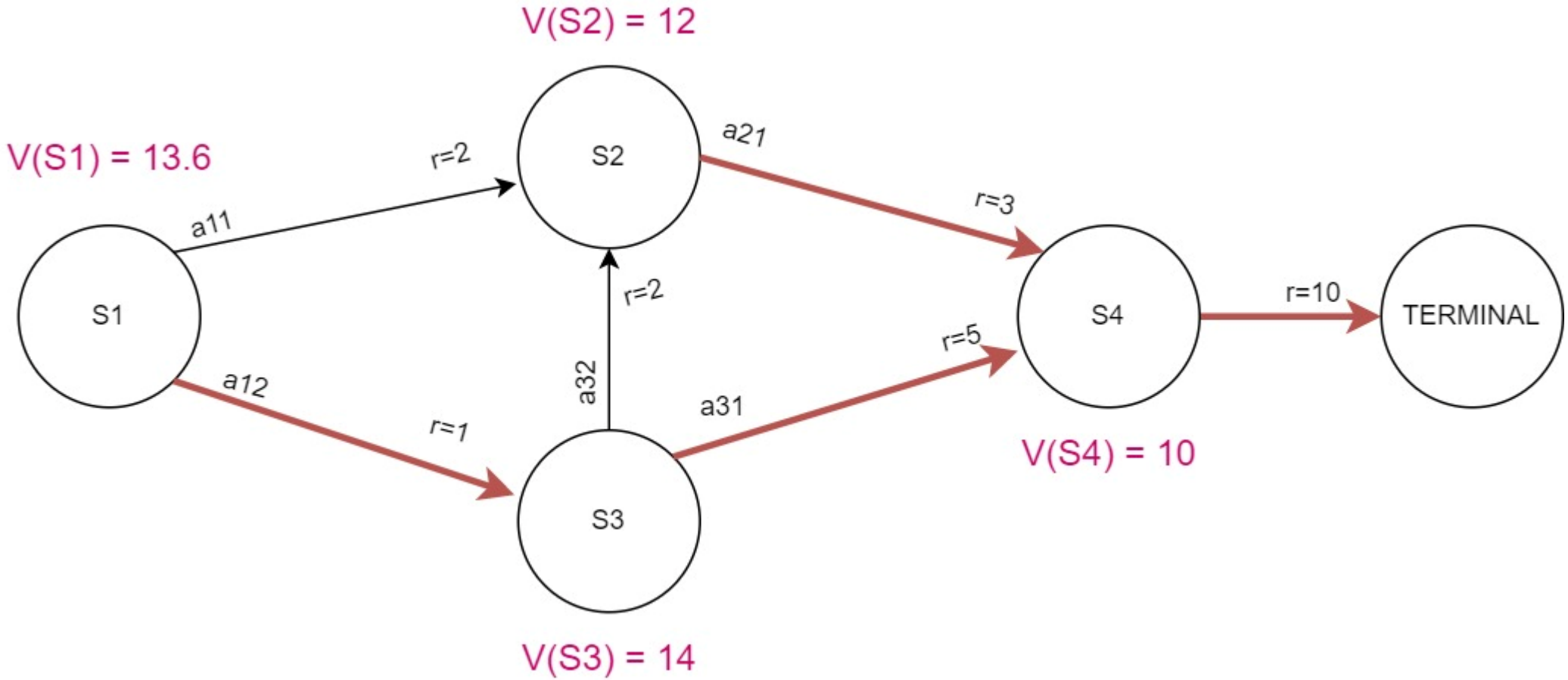

V(S2) = 0
V(S1) = 0
r=2
S2
a21
a11
r=3
S1
r=2
S4
r=10
TERMINAL
a32
r=5
a12
a31
r=1
V(S4) = 0
S3
V(S3) = 0



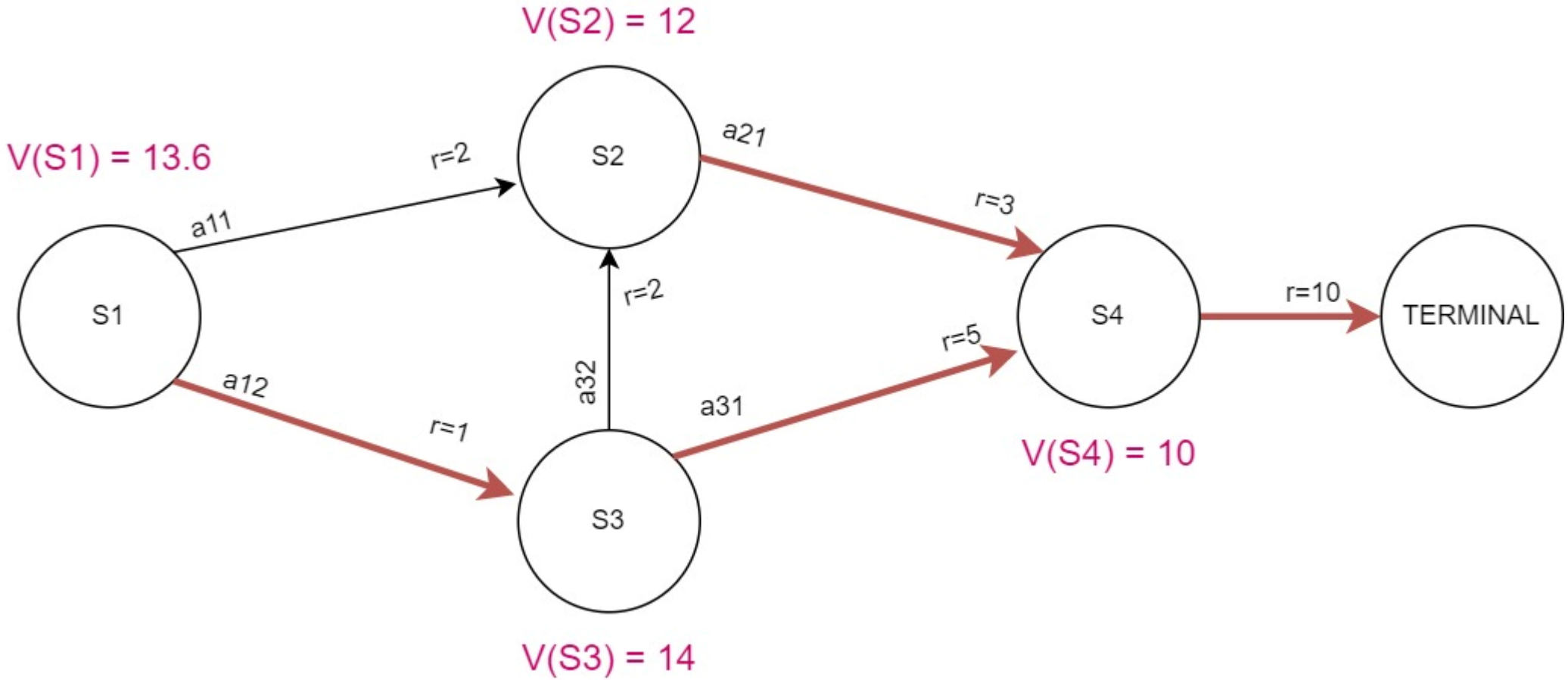
$$V_{k+1}^{\pi_i}(s) \leftarrow \sum_{s'} T(s, \pi_i(s), s') \left[R(s, \pi_i(s), s') + \gamma V_k^{\pi_i}(s') \right]$$



$$\pi_{i+1}(s) = \arg\max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V^{\pi_i}(s') \right]$$



$$V_{k+1}^{\pi_i}(s) \leftarrow \sum_{s'} T(s, \pi_i(s), s') \left[R(s, \pi_i(s), s') + \gamma V_k^{\pi_i}(s') \right]$$



$$\pi_{i+1}(s) = \arg\max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V^{\pi_i}(s') \right]$$

نکات

در value iteration:

- ما فقط v هارا تغییر می دهیم و سیاست به صورت غیر مستقیم تغییر می کند.
- تعیین سیاست بعدا می‌تواند صورت پذیرد، روش معمول انتخاب بهترین عمل است.
  - این کار نرخ اکتشاف را کاهش می دهد اما احتمال عمل بهینه را افزایش می دهد.