that fSSPUDE should be no harder to solve than SSP. This intuition is confirmed by this following result:

**Theorem 3.** $f_{\text{SSPUDE}} = SSP.$

*Proof sketch.* To show that every fSSPUDE MDP $M_{f_{\text{SSPUDE}}}$ can be converted to an SSP MDP, we augment the action set $A$ of fSSPUDE with a special action $a'$ that causes a transition to a goal state with probability 1 and that costs $D$. This MDP is an SSP, since reaching the goal with certainty is possible from every state. At the same time, the optimization criteria of fSSPUDE and SSP clearly yield the same set of optimal policies for it.

To demonstrate that every SSP MDP $M_{\text{SSP}}$ is also an fSSPUDE MDP, for every $M_{\text{SSP}}$ we can construct an equivalent fSSPUDE MDP by setting $D = J^*(s)$. The set of optimal policies of both MDPs will be the same. (Note, however, that the conversion procedure is impractical, since it assumes that we know $J^*(s)$ before solving the MDP.)

The above conversion from fSSPUDE to SSP immediately suggests solving fSSPUDE with modified versions of standard SSP algorithms, as we describe next.

**Value Iteration:** Theorem 3 implies that $J^*$, the optimal cost function of an fSSPUDE MDP, must satisfy the following modified Bellman equation:

$$J(s) = \min \left\{ D, \min_{a \in A} \left[ C(s, a) + \sum_{s' \in S} T(s, a, s') J(s') \right] \right\} \quad (10)$$

Moreover, it tells us that $\pi^*$ of an fSSPUDE must be greedy w.r.t. $J^*$. Thus, an fSSPUDE can be solved with arbitrarily initialized $\text{VI}_{\text{SSP}}$ that uses Equation 10 in updates.

**Heuristic Search:** By the same logic as above, all FIND-AND-REVISE algorithms and their guarantees apply to fSSPUDE MDPs if they use Equation 10 in lieu of Bellman backup. Thus, all heuristic search algorithms for SSP work for fSSPUDE.

We note that, although this theoretical result is new, some existing MDP solvers use Equation 10 implicitly to cope with goal-oriented MDPs that have unavoidable dead ends. One example is the miniGPT package [Bonet and Geffner, 2005]; it allows the user to specify a value $D$ and then uses it to implement Equation 10 in several algorithms including $\text{VI}_{\text{SSP}}$ and LRTDP.

### 6 The Case of an Infinite Penalty

In contrast to fSSPUDE MDPs, no existing algorithm can solve iSSPUDE problems either implicitly or explicitly, so all algorithms for tackling these MDPs that we present in this section are completely novel.

#### 6.1 Value Iteration for iSSPUDE MDPs

As for the finite-penalty case, we begin by deriving a Value Iteration-like algorithm for solving fSSPUDE. Finding a policy satisfying Eq. 9 may seem hard, since we are effectively dealing with a multicriterion optimization problem. Note, however, the optimization criteria are, to a certain degree, independent — we can first find the set of policies whose probability of reaching the goal from $s_0$ is optimal, and then select from them the policy minimizing the expected cost of goal trajectories. This amounts to finding the optimal goal-probability function $P^*$ first, then computing the optimal cost function $[J^*[P^*]]$ conditional on $P^*$, and finally deriving an optimal policy from $[J^*[P^*]]$. We consider these subproblems in order.

**Finding $P^*$.** The task of finding, for every state, the highest probability with which the goal can be reached by any policy in a given goal-oriented MDP has been studied before — it is the MAXPROB problem mentioned in the Background section. Solving a goal-oriented MDP according to the MAXPROB criterion means finding $P^*$ that satisfies

$$P^*(s) = \max_{a \in A} \sum_{s' \in S} T(s, a, s') P^*(s') \forall s \notin G$$

As already discussed, this $P^*$ can be found by the VI$_{\text{MP}}$ algorithm with an arbitrary initializing heuristic.

**Finding $[J^*[P^*]]$.** We could derive optimality equations for calculating $[J^*[P^*]]$ from first principles and then develop an algorithm for solving them. However, instead we present a more intuitive approach. Essentially, given $P^*$, we will build a modification $M_{P^*}$ of the original MDP whose solution is exactly the cost function $[J^*[P^*]]$. $M_{P^*}$ will have no dead ends, have only actions greedy w.r.t. $P^*$, and have a transition function favoring transitions to states with higher probabilities of successfully reaching the goal. Crucially, $M_{P^*}$ will turn out to be an SSP MDP, so we will be able to find $[J^*[P^*]]$ with SSPs’ familiar machinery.

To construct $M_{P^*}$, observe that an optimal policy $\pi^*$ for an iSSPUDE MDP, one whose cost function is $[J^*[P^*]]$, must necessarily use only actions greedy w.r.t. $P^*$, i.e., those maximizing the right-hand side of Eq. 11. For each state $s$, denote the set of such actions as $A^*_s$. We focus on non-dead ends, because for dead ends $[J^*[P^*]](s) = 0$, and they will not be part of $M_{P^*}$. By Eq. 11, for each such $s$, each $a^* \in A^*_s$ satisfies $P^*(s) = \sum_{s' \in S} T(s, a^*, s') P^*(s')$. Note that this equality expresses the following relationship between event probabilities:

$$P \left( \text{Goal was reached from } s \text{ via optimal policy} \right)$$

$$= \sum_{s' \in S} P \left( s \rightarrow s' \text{ transition } \wedge \text{ Goal was reached from } s \text{ via optimal policy} \right),$$

or, in a slightly rewritten form,

$$\sum_{s' \in S} P \left( a^* \text{ caused } s \rightarrow s' \text{ transition } \mid \text{ Goal was reached from } s \text{ via optimal policy} \right) = 1,$$