Weighted Programs and Ethical Planning

Ondrej Majer and Igor Sedlár

The Czech Academy of Sciences, Institute of Computer Science
Prague, The Czech Republic

1 Ethical planning

In standard planning theory [6], a plan is a sequence of actions. A planning problem can be seen as a transition system with a starting state, a designated set of goal states, and a function assigning a relation on states of the transition system to each action. A plan solves a planning problem if the given sequence of actions is guaranteed to reach the set of goal states.

One may compare plans from many points of view, including resource consumption etc. and this comparison is relevant when it comes to plan generation. One aspect of plans that may (and should) be taken into account in addition to reaching the planning goal is the *ethical impact* of the (potential) execution of the plan. How much damage will be done by executing the given plan? Is there a plan that will reach the goal but will result in less damage?

Devising formal models for reasoning about ethical planning is an interesting research topic (see e.g. [1], [9]). In a recent paper, Grandi et al. [7] have developed an approach based on a combination of linear temporal logic with lexicographic preference modelling. In our contribution, we will complement the previous research on the topic by developing an approach based on *weighted programs* [3].

2 Weighted programs

Weighted programs [3] are an extension of standard while programs (featuring assignments of values to variables, sequential composition, conditional if-then-else statements and while loops) with non-deterministic choice and an operation of adding weight to the current execution trace. We will work with a propositional abstraction of weighted programs where atomic programs are unstructured.

Definition 1 (Weighted programs). *Fix three disjoint sets of variables*

- *Statements:* $B, C := \mathtt{b} \mid \neg B \mid B \wedge C \mid B \vee C$
- *Programs:* $P, Q := \mathtt{p} \mid P; Q \mid \textbf{if } B \textbf{ then } P \textbf{ else } Q \mid \textbf{while } B \textbf{ do } P \mid P \oplus Q$
- *Weightings:* $E, F := \mathtt{e} \mid 1 \mid 0 \mid E \cdot F \mid E + F$

The program $P \oplus Q$ chooses between executing program P and executing program Q non-deterministically. Weightings are expressions of the semiring signature; this reflects the general approach to weights in [3] where the particular structure of the algebra of weights that fits a particular application (e.g. the tropical semiring) is not imposed in general. However, semirings do capture some general aspects of reasoning about weights, including the notions of "no weight" (1), "absolute weight" (0), weight addition ($\cdot$) and weight minimization ($+$). Intuitively, a weighting E is a special kind of program that adds the value corresponding to the semiring expression E to the accumulated weight of the given execution trace of a program. It is argued in [3] that weighted programs provide a flexible framework for encoding mathematical models, such as optimization problems or probability distributions, in terms of an algorithmic representation.

Example 1. An example of a weighted program is

$$\textbf{while } \neg \mathtt{b} \textbf{ do } (\mathtt{p} \otimes \mathtt{q}); (\textbf{if } \mathtt{c} \textbf{ then } \mathtt{e} \textbf{ else } 1)$$

Intuitively, what this program does is that while the statement `b` is false, it non-deterministically chooses between doing `p` or doing `q` after which the condition `c` is tested; if `c` is true, that the weight `e` is added to the current execution trace and if `c` is false, no weight is added.

Definition 2 (Model). *A weighted program model is a pair $M = (W, \boldsymbol{S}, V)$, where $W \neq \emptyset$ is a set (of worlds), $\boldsymbol{S}$ is a semiring, and V is a function assigning to each statement variable a subset of W, to every program variable a binary relation on W, and to every weighting variable an element of $\boldsymbol{S}$.*

The function V is lifted to all statements and weightings in the expected way, and to all programs in the manner usual in operational semantics [2], with the proviso that $V(P \oplus Q) = V(P) \cup V(Q)$.

3 Programs, plans, and ethical planning

Planning problems are represented by triples (M, w, G) where $w \in W$ and $G \subseteq W$. It is useful to represent G by a statement variable `goal`. A program P solves a planning problem (M, w, G) iff $V(P)(w) \subseteq G$.

A *trace* is a sequence of program variables and weighting expressions. Each program P generates a set of traces $Tr(P)$ which can be defined by induction on the structure of P in the obvious way. Given any pair (M, w), the set of traces $Tr_{(M,w)}(P)$ is the set of traces generated by P when executed in state w of the model M. Each trace t corresponds to a *plan* $\mathbf{p}(t)$ that is obtained by erasing all weighting expressions from t. Given each trace t, the *weight of* t is $\mathbf{w}(t)$ is obtained by erasing all program variables from t. For convenience, we will not distinguish between a sequence $E_1 \ldots E_n$ of weighting expressions and the expression $E_1 \cdot \ldots \cdot E_n$. Given any model M and trace t, $V(\mathbf{w}(t)) \in \boldsymbol{S}$ is the accumulated weight of the trace t. For a set of traces T, we may define $\mathbf{w}(T) = \{\mathbf{w}(t) \mid t \in T\}$. If $\boldsymbol{S}$ is complete, then we may define $V(\mathbf{w}(T)) = \sum_{t \in T} V(\mathbf{w}(t))$. Using complete idempotent semirings, $V(\mathbf{w}(T))$ is the weight of the optimal run of any program that generates T when run in some (M, w). We define $V(\mathbf{w}(P))$ in an obvious way; this is the optimal run of program P in the model at hand.

Batz et al. [3] develop a Dijkstra-style weakest precondition calculus that allows to compute (to put it in our terminology) the optimal weight $V(\mathbf{w}(P))$ recursively on the structure of P.

Weighted programs provide a framework for reasoning about ethical planning via plan-generating programs as follows. Given any model M, the *ethical weight* of a statement B can be encoded via the instruction

$$\textbf{if } B \textbf{ then } E \textbf{ else } F$$

stating that if B is true, then weight E is added to the current execution trace and otherwise weight F is added (often we want $F = 1$, meaning that no weight is added if B is false). If $B_1, \ldots, B_n$ are "ethically sensitive" statements (expressing states of affairs that, when they occur, they constitute some form of damage) with weights $E_1, \ldots, E_n$, then

$$ET := \textbf{if } B_1 \textbf{ then } E_1 \textbf{ else } F_1; \ldots; \textbf{if } B_n \textbf{ then } E_n \textbf{ else } F_n$$

constitutes an *ethical test* of the current state of execution of a program. An "ethically sensitive" plan-generating program can then be devised by inserting ET after every instruction. The semiring element $V(\mathbf{w}(P))$ for such program P then expresses the value of "ethically optimal" runs of the program P.

4 Kleene algebras for ethical planning

Kleene algebras with tests [8] constitute a simple algebraic framework for reasoning about properties of while programs. An extension of Kleene algebra with tests that models reasoning about weighted programs was introduced in [10]. Kleene algebra with domain [4, 5] extends Kleene algebra with tests with a weakest precondition-style operator. In the final part of the presentation, we will discuss how a combination of Kleene algebra with weights and tests of [10] with Kleene algebra with domain formalizes reasoning about "ethically optimal" runs of plan-generating weighted programs.

References

[1] S. Alili, R. Alami, and V. Montreuil. A task planner for an autonomous social robot. *Distributed autonomous robotic systems 8*, pages 335–344, 2009.

[2] K. R. Apt, F. S. de Boer, and E.-R. Olderog. *Verification of Sequential and Concurrent Programs.* Texts in Computer Science. Springer, 3rd edition, 2009.

[3] K. Batz, A. Gallus, B. L. Kaminski, J.-P. Katoen, and T. Winkler. Weighted programming: A programming paradigm for specifying mathematical models. *Proc. ACM Program. Lang.*, 6(OOPSLA1), apr 2022.

[4] J. Desharnais, B. Möller, and G. Struth. Kleene algebra with domain. *ACM Trans. Comput. Logic*, 7(4):798–833, oct 2006.

[5] J. Desharnais and G. Struth. Internal axioms for domain semirings. *Science of Computer Programming*, 76(3):181–203, 2011. Special issue on the Mathematics of Program Construction (MPC 2008).

[6] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning: Theory and Practice.* Elsevier, 2008.

[7] U. Grandi, E. Lorini, T. Parker, and R. Alami. Logic-based ethical planning. In *AIxIA 2022 – Advances in Artificial Intelligence*, pages 198–211. Springer International Publishing, 2023.

[8] D. Kozen. Kleene algebra with tests. *ACM Trans. Program. Lang. Syst.*, 19(3):427–443, May 1997.

[9] E. Lorini. A logic for reasoning about moral agents. *Logique et Analyse*, pages 177–218, 2015.

[10] I. Sedlár. Kleene algebra with tests for weighted programs. In *ISMVL 2023*, 2023.