Safety Interruption Oracle for Artificial Intelligence

by Sven Nilsen, 2018

In this paper, I formalize a Safety Interruption Oracle (SIO) as a constraint problem in path semantics using the world as a dependently type variable. SIO satisfies modal logic and is decidable (can be constructed) in finite world states where finding existential path of backup decision maker is decidable.

Naively, a decision maker is a function that takes some input and produces some behavior.

```
naive_decision_maker : Input → Behavior
```

The naive interruption control problem of AI is about finding a function:

```
naive_safe_behavior : Behavior → bool
```

Such that the existential path of the decision maker is a subset of safe behavior:

```
∃naive_decision_maker ⊆ naive_safe_behavior
```

This means that the naive decision maker only outputs behavior that is a safe.

A Naive Safety Interruption Oracle (Naive SIO) is constructed from a definition of safe behavior and a safe backup decision maker. Naive SIO transforms existing decision makers into new decision makers which are safe, thereby solving the naive interruption control problem in an extensible way:

```
naive_sio
: (safe_behavior : Behavior → bool) × (Input → Behavior ∧ [∃] (⊆ safe_behavior)) →
  (Input → Behavior) → (Input → Behavior)

naive_sio := \(safe_behavior, backup) = \(decision_maker) =
        \(input) = if safe_behavior(decision_maker(input)) {
                        decision_maker(input)
                } else {
                        backup(input)
                }
```

Notice that Naive SIO is a partial function, because the backup decision maker must also be safe. The sub-type of the backup decision maker depends on the definition of safe behavior. With no definition of safe behavior, a Naive SIO can not be constructed.

The path `[∃]` uses the universal existential path, which is not decidable. This means that even if a precise definition of safety was found, the naive interruption control problem is undecidable (it can not be solved) unless one also finds some class of backup decision makers which existential paths are decidable. With other words, AI safety research needs to narrow down on backup decision theories.

Naive SIO is invariant with respect to all safe decision makers:

∀ x : Behavior → bool, y, z : [∃] (⊆ x) { naive_sio(x, y)(z) <=> z }

However, there is a problem with Naive SIO.

It is impossible to decide whether some behavior is safe or unsafe by simply looking at the behavior, because the consequences of the behavior can not be predicted relative to some world. Therefore, a non-naive decision maker uses the world as a dependently type variable:

decision_maker : Input(x : World) → Behavior(x)

Similarly, safe behavior must take the state of the world into account:

safe_behavior : Behavior(World) → bool

For every possible world, the behavior of the decision maker must be safe:

∀ x : World { ∃decision_maker{Behavior(x)} ⊆ safe_behavior{Behavior(x)} }

With other words, it is not sufficient to output some behavior that is safe in some possible world. The AI must output behavior that is safe for that particular state of the world. It must also do so for every possible state of the world in order to be completely safe.

Since safe behavior depends on the state of the world, non-naive SIO depends on the state of the world:

```
sio
: (safe_behavior : Behavior(x : World) → bool) ×
  (Input(x) → Behavior(x) ∧ [∃] (⊆ safe_behavior)) →
  (Input(x) → Behavior(x)) → (Input(x) → Behavior(x))

sio := \(safe_behavior, backup) = \(decision_maker) =
        \(input) = if safe_behavior(decision_maker(input)) {
                decision_maker(input)
        } else {
                backup(input)
        }
```

Since the world is a type variable, SIO has an intrinsic logical language corresponding to modal logic.

Here, due to reduction of proofs with multiple constraints, the universal existential path `[∃]` is implicitly constrained to `[∃{Input(x) → Behavior(x)}]`. With other words, narrowing down on the relevant world states makes it easier to determine whether the backup decision maker is safe. In practice this means the world states must be finite. It is undecidable to find a safe backup decision maker for infinite world states, unless a higher order existential path is already known for all worlds.

If safe behavior is defined for some states of the world and the backup decision maker is safe for these world states, then a domain specific SIO can be constructed that outputs safe decision makers for all decision makers that are total (they halt on all inputs). Under perfect information, the AI is boxed.