Adversarial Path of Cartesian Product

by Sven Nilsen, 2018

In this paper I represent the adversarial path of Cartesian products.
This is based on ideas from a discussion with Adam Nemecek.

A Cartesian product is just a tuple:

```
a : A
b : B
(a, b) : A × B
```

The paper “Adversarial Paths” introduced choices and adversarial choices.
Making a choice `A` is written `A ~ 0`, which has the type:

```
A ~ 0 : T → A ~ 1
```

A Cartesian product of choices is itself a choice.
In the case of a making a choice of a Cartesian product of choices:

```
(a, b) ~ 0 <=> (a ~ 0, b ~ 0)
```

For example, when `a` and `b` are lists:

```
(a, b) : [A] × [B]
(a, b) ~ 0 : ([A] × [B]) ~ 0
(a ~ 0, b ~ 0) : [A] ~ 0 × [B] ~ 0
(a ~ 0, b ~ 0) : (nat → A ~ 1) × (nat → B ~ 1)
(a ~ 0, b ~ 0) : nat × nat → A ~ 1 × B ~ 1
(a ~ 0, b ~ 0) : nat × nat → (A × B) ~ 1
```

Therefore:

```
(a, b) ~ 0 : nat × nat → (A × B) ~ 1
```

```
a : [A]
b : [B]
```

A such tuple of list of choices is called “undecidable” because it lacks the necessary information to arrive at any concrete new resource `(A × B) ~ 1`. To do this, one has to apply the tuple to arguments:

```
(a, b)(x, y) : (A × B) ~ 1          `(a, b)(x, y)` is “decidable”
```

```
x, y : nat
```