1 Code-breaking Games

1.1 Notation

Let Form_X be the set of all prepositional formulas over the set of variables X; V_X be the set of all valuations of variables X. Formulas $\varphi_0, \varphi_1 \in \mathrm{Form}_X$ are (semantically) equivalent, written $\varphi_0 \equiv \varphi_1$, if $v(\varphi_0) = v(\varphi_1)$ for all $v \in V_X$. For a formula $\varphi \in \mathrm{Form}_X$, let $\tau_X(\varphi) = |\{v \in V_X \mid v(\varphi) = 1\}|$ be the number of valuations by which φ is satisfied. We often omit the index X if it is clear from the context.

For any unary predicate P, $\#i \in A.P(i) = |\{i \in A \mid P(i)\}|$. We often omit the "$\in A$" part and write only $\#i.P(i)$ if the range of i is clear from the context.

Let Perm_X be the set of all permutations of X.

1.2 Formal definition

Definition 1. A *code-breaking game* is a quintuple $\mathcal{G} = (X, \varphi_0, T, E, \Phi)$, where

- X is a finite set of propositional variables,
- $\varphi_0 \in \mathrm{Form}_X$ is a satisfiable prepositional formula,
- T is a finite set of types of experiments,
- $E \subseteq T \times X^\star$ is an *experiment* relation, and
- $\Phi : E \to 2^{\mathrm{Form}_X}$ is an *outcome function* such that $\Phi(e)$ is finite for any $e \in E$, (and for $\Phi(e) = \{\psi_1, \ldots, \psi_k\}$, it holds

  $$\forall v \in V_X : v(\varphi_0) = 1 \Rightarrow \exists \psi \in \Phi(e) \,.\, v(\psi) = 1$$

  i.e. $\varphi_0 \Rightarrow \psi_1 \vee \psi_2 \vee \ldots \vee \psi_k$ is true.

Intuitively, the objective of the game is to find a valuation of variables X by a series of experiments. Let us call it *the wanted valuation*. The search space is reduced by formula φ_0, which is always known to be satisfied by the wanted valuation.

Experiments consist of a type, which is from the set T and a parametrization, which is a string of variables from X. The experiment relation E specifies all permitted parameterizations for each type of experiment and, therefore, E is the set of all possible experiments as pairs.

The outcome function gives us the possible outcomes of an experiment. We require that at least one of them must be satisfied by every valuation by which φ_0 is satisfied.

Example 2 (Fake-coin problem). Fake-coin problem with n coins, one of which is fake, can be formalized as a code breaking game $\mathcal{F}_n = (X, \varphi_0, T, E, \Phi)$, where

- $X = \{x_1, x_2, \dots, x_n, y\}$
  Intuitively, variable x_i tells weather the coin i is fake. Variable y tells weather it's lighter or heavier.
- $\varphi_0 = \text{Exactly-1}\left(\{x_1, \dots, x_n\}\right)$
  This is to ensure that exactly one coin is fake.
- $T = \{t\}$
  There is only one type of experiment – weighting the coins.
- $E = \{(t, p) \mid p \in \{x_1, \dots, x_n\}^{2n}, n \geq 0, \forall x \in X : \#_x(p) \leq 1\}$
  Any sequence of variables of even length with no repetitions is a permitted parametrization of type t.
- $$\begin{aligned}\Phi((t,p)) = \big\{&(\bigvee A \wedge \neg y) \vee (\bigvee B \wedge y),\\ &(\bigvee A \wedge y) \vee (\bigvee B \wedge \neg y),\\ &\neg \bigvee (A \cup B)\big\},\end{aligned}$$

  where $A = \{p[i] \mid 1 \leq i \leq |p|/2\}$, $B = \{p[i] \mid |p|/2 < i \leq |p|\}$.

Example 3 (Mastermind). Mastermind puzzle with n pegs and color set C can be formalized as a code breaking game $\mathcal{M}_{n,C} = (X, \varphi_0, T, E, \Phi)$, where

- $X = \{x_{i,c} \mid 1 \leq i \leq n, c \in C\}$.
  Variable $x_{i,c}$ tells whether there is the color c at position i. For simplicity, let us use the notation $X_c = \{x_{i,c} \mid 1 \leq i \leq n\}$.
- $\varphi_0 = \bigwedge \{\text{Exactly-1}\, \{x_{i,c} \mid c \in C\} \mid 1 \leq i \leq n\}$.
  This guarantees that there is exactly one color at each position.
- $T = \{t\}$.
  There is only one type of experiment – guessing a combination.
- $E = \{(t, p) \mid p = x_{1,c_1} x_{2,c_2} \dots x_{n,c_n}\}$.
  Parametrization of t can be any string of length n, i-th symbol of which belongs to $\{x_{i,c} \mid c \in C\}$.

- Inference function is defined by

$$\begin{aligned}\Phi(v,(t,p)) = {} & \text{Exactly-b}\,\{p[i] \mid 1 \le i \le n\} \land \\ & \text{Exactly-t} \bigcup \big\{ \\ & \{\text{AtLeast-k}\,\{x_{i,c} \mid 1 \le i \le n\} \mid 1 \le k \le \#i.(p[i] \in X_c)\} \\ & \mid c \in C \big\}\end{aligned}$$

  where $b = \#i.(v(p[i]) = 1)$ captures the number of black pegs in the response for the experiment (t,p) and $t = \sum_{c\in C} \min(\#i.(v(x_{i,c}) = 1), \#i.(p[i] \in X_c))$ is the total number of pegs (black + white).
  Fakt to nejde nějak jednodušej?

1.3 Strategies

Definition 4. A *strategy* is a function $\sigma : \mathrm{Form}_X \to E$, determining the next experiment for given accumulated knowledge, such that

$$\varphi_0 \equiv \varphi_1 \Rightarrow \sigma(\varphi_0) = \sigma(\varphi_1).$$

A strategy σ together with a secret valuation v induce a *solving process*, which is an infinite sequence

$$\pi_{\sigma,v} = \varphi_0 \xrightarrow{e_1} \varphi_1 \xrightarrow{e_2} \varphi_2 \xrightarrow{e_3} \dots$$

such that $e_{i+1} = \sigma(\varphi_0 \land \varphi_1 \land \dots \land \varphi_i)$ and $\varphi_{i+1} = \Phi(v, e_{i+1})$ for all $i \in \mathbb{N}_0$. For the sake of simplicity, let us write $\varphi_{0..k}$ instead of $\varphi_0 \land \varphi_1 \land \dots \land \varphi_k$.
We define *length* of the solving process, denoted $|\pi_{\sigma,v}|$ (despite the infinite length of the sequence), as the smallest $k \in \mathbb{N}_0$ such that $\tau_X(\varphi_{0..k}) = 1$. This corresponds to the situation in which we can unambiguously determine the secret code.
Note that it always holds $\tau(\varphi_{0..k}) > 0$ because $v(\varphi_{0..k}) = 1$ thanks to the condition (i) in Definition 1.
The following lemma is a straightforward consequence of the memory-less nature of the games. It says that once a strategy gives us an experiment that yields no new information, we will never more get any new information (using the strategy).

Lemma 5. *If $\tau(\varphi_{0..k}) = \tau(\varphi_{0..k+1})$ for some $k \in \mathbb{N}$, then $\tau(\varphi_{0..k}) = \tau(\varphi_{0..k+l})$ for any $l \in \mathbb{N}$.*

Proof. If $\varphi_{0..k+1} = \varphi_{0..k} \land \varphi_{k+1}$ is satisfied by valuation v, so must be $\varphi_{0..k}$. Since $\tau(\varphi_{0..k}) = \tau(\varphi_{0..k+1})$, the sets of valuations satisfying $\varphi_{0..k}$ and $\varphi_{0..k+1}$ must be exactly the same and the formulas are thus equivalent. This implies $\sigma(\varphi_{0..k}) = \sigma(\varphi_{0..k+1})$ and thus also $\varphi_{k+2} = \varphi_{k+1}$. By induction, $\varphi_{k+l} = \varphi_{k+1}$ and $\varphi_{0..k+l} \equiv \varphi_{0..k}$ for any $l \in \mathbb{N}$. ■

The *worst-case number of experiments* λ^σ of a strategy σ is the maximal length of the solving process $\pi_{\sigma,v}$ over all valuations v, i.e. $\lambda^\sigma = \max_{v \in V_X} |\pi_{\sigma,v}|$. We say that the strategy *solves the game* if λ^σ is finite. The game is *solvable* if there exists a strategy that solves the game.

Problem 6. *Given a code-breaking game $\mathcal{G}$, decide whether $\mathcal{G}$ is solvable.*

Definition 7. A strategy σ is *optimal* if $\lambda^\sigma \leq \lambda^{\sigma'}$ for any strategy σ'.
A strategy σ is *greedy* if for every $\varphi \in \mathrm{Form}_X$ and $e' \in E$,

$$\max_{v \in V_X} \tau_X(\varphi \wedge \Phi(v, \sigma(\varphi))) \leq \max_{v \in V_X} \tau_X(\varphi \wedge \Phi(v, e')).$$

In words, a greedy strategy minimizes the worst-case number of possible valuations in the next step.

Problem 8. *Given a code-breaking game $\mathcal{G}$, decide whether all greedy strategies are optimal. This seems to be the case for Fake-coin problem (?) but it is not the case for Mastermind[ref].*

Bibliography