Petri Sport: A Sport for Petri Netters

Stefan Klikovits, Alban Linard, Dimitri Racordon, and Didier Buchs *

Software Modeling and Verification (SMV) Group, Faculty of Science
University of Geneva Geneva, Switzerland
`firstname.lastname@unige.ch`

Abstract. Petri nets are a family of formalisms dedicated to the representation of concurrent systems. Their strength is the compact modeling of complex behaviors using very simple rules. Despite this simplicity, many teachers observe that students often require a lot of exposure and numerous exercises to truly understand the semantics of Petri nets. In order to speed up this learning process and provide a different attack angle, we propose Petri sport, a fun game based upon the Petri net formalism. In Petri sport, players aim to gather points by moving across a Petri net-shaped playing field and "firing" transitions. A clock-based play style supports a structured game advance while at the same time it encourages players to move fast. As the playing field is shaped like a Petri net, it is possible to challenge a player's movement speed, intellectual capabilities, as well as team coordination and communication. The difficulty level of Petri sport is based on the choice of playing field. This allows for adaptation in order to best fit the competitors' age, experience and/or physical fitness level.

1 Introduction

A significant part of the system and software engineering community embraces Petri nets for their capability to model and simulate complex behaviors, such as concurrent systems. The adoption of Petri nets can be explained by their clear and elegant semantics, as well as their simple graphical notation. Since their introduction [Pet62], there have been many efforts to extend, adapt and modify the formalism. Such efforts materialized for instance in the addition of inhibitor arcs [Zai14], which provide Turing completeness, or Colored Petri nets [Jen14], which enable for more compact representations, by adding the possibility to use distinguishable types of tokens.

These concepts are useful and make the Petri net formalism (family) powerful and versatile, albeit difficult to learn. In the course of our teaching activities, we regularly observe that a significant part of our students struggle to fully understand the Petri net semantics. As a result, confusion and misunderstandings

* This project is supported by: FNRS STRATOS : Strategy based Term Rewriting for Analysis and Testing Of Software, the Hasler Foundation, 1604 CPS-Move and COST IC1404: MPM4CPS

slow down the learning process and hinder the understanding of advanced Petri net variants and tooling.

This motivated us to tackle the problem with a different approach, which led us to the design of Petri sport. Petri sport is a game based on the Petri net formalism. The sport borrows from the graphical syntax of the formalism to design playing fields, and from its semantics to establish the playing rules. The game's concept originated as a means for students unfamiliar with Petri nets to observe its semantics by actively "being part of it". Through further gamification [Kap12], and in particular the introduction of a competitive aspect, Petri sport became a canvas which can be used to teach advanced Petri net concepts through the lens of an entertaining activity.

We designed a handful of different Petri sport variants so that the game's difficulty is adaptable to the player's theoretic knowledge, experience and physical fitness. While in its most basic form the game underlines coordination, quick thinking and stamina, we also developed puzzle modes and similar variants, which aim at challenging players' intellects in a less physical way. Based on our observations of Petri sport's entertainment potential, we are certain that even senior researchers will enjoy this "academic" sports discipline[1]. The rest of the paper is structured as follows: Section 2 outlines the rules of Petri sport and explains the playing field, the teams and the game play. Section 3 discusses the formal basis which underlies Petri sport. Section 4 discusses different game variants and adaptations that can be incorporated. Section 5 elaborates on the constraints of the playing fields and properties that should be satisfied in order to guarantee entertaining game rounds. Section 6 provides details about the experiments we performed. Section 7 discusses related work and similar approaches. Section 8 concludes.

2 Rules

Petri sport is a team-based game, played on a field whose shape resembles a Petri net. Each team consists of several players who act as the tokens of the Petri net. At the beginning of the game, players are either within places, in which case they are called *bound* players, or alongside the playing field, in which case they are called *free* players. Similarly to Petri nets, we use the term *marking* to refer to an assignment of players (i.e. tokens) to places. To advance the game, players must *fire* transitions, according to a set of rules describing how tokens are consumed and produced.

In order to fire a particular transition, *bound* players must move from precondition places to the transition, following the arcs of the laid out Petri net. Once they reach the transition, they meet *free* players who then carry on the firing process by moving to the postcondition places. Bound players satisfying the preconditions become free only after the transition firing completed, i.e. when the formerly free players reached their respective postcondition places. On the

[1] We believe that Petri sport might even bring sustainable health benefits to the Petri net community.

Fig. 1: A group of students playing Petri sports

other hand, free players satisfying the postconditions become bound upon completion of a transition firing. Hence, the total number of bound and free players involved depends on the *weight* of the input and output arcs of the transition, respectively. As in a classic Petri net, a transition can only be fired when it is *enabled*. This requires that the number of players in the preconditions at least matches the weight annotation of the arc coming out of these preconditions. In addition, the number of free players available should be at least the sum of the postcondition weights. Figure 2 shows examples of the *enable* semantics.

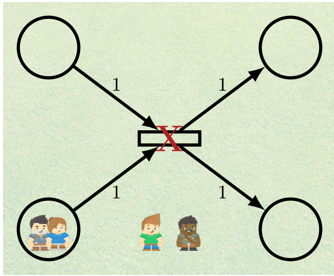
Conceptually, the firing semantics can be represented as a classic Petri net where each transition is connected to a special *free* place by pre- and postcondition. *free* is the virtual place where free players are positioned before/after they are bound. The weight of the arc from *free* to the transition is the sum of postcondition weights, while the weight of arc from the transition to *free* is the sum of the precondition weights. An example is depicted in Figure 3.

This behavior helps avoiding the common misconception that tokens are *passing* from preconditions to postconditions. Our semantics clearly shows that players who are freed from preconditions are not the same as the players who get bound to postconditions.

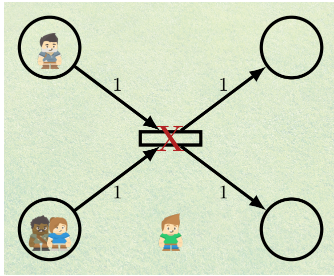
Petri sport rules revolve around the concept of correctly firing transitions. The overall goal is to fire certain point-awarding transitions as often as possible within limited time. Depending on the "game flavor", rules may be added or modified in order to make the game more entertaining and/or challenging. A discussion of such variants is given in Section 4.

2.1 Field

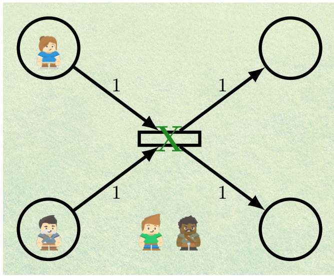
A typical Petri sport field (a *sport Petri net, SpPN*) consists of places and transitions which are distributed and connected via arcs. Places are traditionally



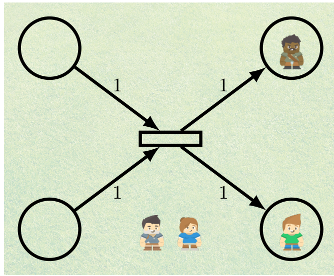
(a) Precondition violation



(b) Not enough free players



(c) Enabled transition



(d) State after firing Figure 2c

Fig. 2: Examples of enabled or disabled transitions with a team of four players. Transitions in Figure 2a and Figure 2b are disabled since there are not enough *bound* players in precondition places, and not enough *free* players, respectively. Figure 2c shows an enabled transition. Firing this transition leads to the marking shown in Figure 2d. Note that the bound players became free and the previously free players are now bound. *We thank* `http://kenney.nl/assets` *for the sprites.*

represented as circles, while transitions are marked as rectangles. We found that hula hoops and small fitness carpets nicely lend themselves to represent places and transitions, but any other means is thinkable, as long as places and transitions are easy to see and distinguishable from each other. Arc connections are drawn lines or strings (tent pegs are useful to tie them to the ground). It is important to annotate arcs with their respective weight (if greater than 1). In our experience sheets of paper with large, bold, well-legible numbers usually suffice.

Petri sport can be played on any flat terrain both indoors and outdoors, provided there is enough space and the risk of injury is reasonably low. The field's size is not fixed and can vary depending on the teams' sizes, players' familiarity with both Petri nets and Petri sport, and players' physical fitness.

Most of the above descriptions are suggestions and should be adapted by the game hosts. The authors appeal to the game organizers' creativity to find the best means possible to establish a playing field. For example, if playing on a

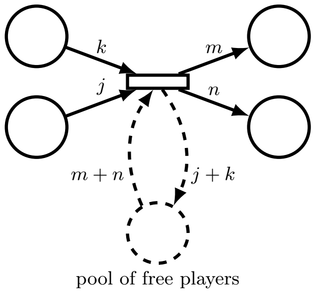


Fig. 3: Representation of the firing semantics of Petri sport

concrete surface it is quick, easy and cheap to simply draw a field using chalk, instead of using hoops, mats and strings as on a grass pitch.

2.2 Example sport Petri net

An example of Sport Petri net (SpPN) is given in Figure 4. The SpPN consists of six places (p_0 – p_5) and eight transitions (t_0 – t_7). Players aim to fire the point-winning transition t_2 as often as possible.

The initial marking displayed in Figure 4, prescribes two players to be placed in p_5. From there, there are two main strategies to follow in order to reach t_2. Players can follow the sequence $t_5 \rightarrow t_6 \rightarrow t_3 \rightarrow t_4 \rightarrow t_2$ which leads to scoring two points and resetting the net to its initial state. Alternatively, a player can reach t_2 by firing $t_7 \rightarrow t_1 \rightarrow t_2$. Assuming all tokens produced by t_1 advance to fire t_2 immediately, this sequence awards five points but involves much longer transitions (in terms of distance). Further, after this transition there are five tokens bound within the SpPN.

This SpPN also features a *sink transition* t_0 which frees two tokens bound within p_0 and p_1, respectively. Deadlocks can occur when too many players are in places p_0 and p_1, as there can be not enough free tokens for the postcondition of t_1. t_0 can then be used to provide these free players. Its firing requires coordination between players in p_0 and p_1. However, it is important to understand that this can easily lead to deadlock situation, such as when firing the sequence $t_7 \rightarrow t_0 \rightarrow t_7 \rightarrow t_0$ since there are no more bound tokens on the field.

2.3 Teams

Petri sport is a team-sport. The minimum number of teams is one, but in our experience a competitive play between at least two even-sized teams increases the enjoyment of the game. At any given moment only one team is actually playing. This means that the teams alternate between rounds and try to score a maximum of points in their respective round – there is no direct engagement between the teams.

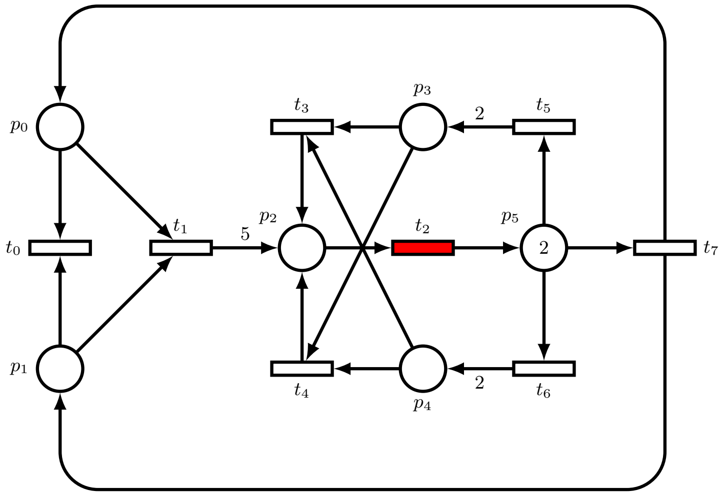


Fig. 4: An example sport Petri net. Intended for teams of ten players, this playing field awards points for firing transition t_2.

The size of a team heavily depends on the playing field. While theoretically the number of players is unlimited, a too large number of players often leads to too large number of enabled transitions, confusing situations and unclear decisions. From our experience, a team size of around 8-12 players is ideal, as this provides many situations where trade-offs have to be taken into consideration and a strategy has to be devised in order to avoid reaching deadlocks.

2.4 Game setup

During the setup phase players can get an overview of the playing field and briefly discuss strategies. Subsequently, the players spread across the field according to the *initial marking*, that is the players-to-places assignment that has to be upheld at the start of the game. It must be provided by the game host and defined together with the playing field.

Players that are not assigned to any places become free and position themselves around the playing field. It is important that free players leave the immediate playing field (it should be easy to see when a player is inside or outside the pitch), in order to clearly distinguish them from bound ones.

This setup phase takes, depending on the field's complexity, between one and two minutes. Immediately at the end of this timespan the main phase is started without any pause in between.

2.5 Main phase

The main game phase is dictated by a clock providing a constant pace. In regular intervals (e.g. every 30 seconds) the clock emits a loud signal (a *beep*). Transition firings have to start clearly after one beep and finish before the next one. The completion of a firing brings the marking to a “normal” state. This means that the players that were freed leave the immediate playing area (i.e. outside of the field) and the newly bound players are positioned clearly within their new places. Teams are awarded points for successfully firing certain predefined transitions. In the example in Figure 4 transition t_2 is the only point-scoring transition, but in general there might be more than one.

Sequential firings Following the firing of one transition, and thereby the freeing of bound and production of newly bound players, it is possible for players to be involved in another transition before the end of the clock interval (i.e. before the next beep). Requirement for the firing of another transition is that the previous firing is completed and all players involved in the new transition have reached their final positions of the previous one (free tokens outside the field, bound tokens reached their post-places). It is thus possible for a team to fire multiple transitions in sequence before a *beep* occurs.

Parallel firings In addition players are also allowed to fire several transitions in parallel. Instead of chaining transitions, parallel firing is performed by firing several transitions concurrently. It requires enough free players for the postconditions of all the transitions, and is thus less likely to happen than sequential firings.

Transaction All transitions that are fired during one clock period are collectively referred to as *a transaction*. Figure 5 shows a transaction with two parallel sequences $S_1 = t_5 \to t_6 \to t_4$ and $S_2 = t_7 \to t_1$. The transition firings within those sequences can be started at any point between two beeps, but have to finish before the next beep. Transition t_1 in this example would not finish firing before the end of the interval and thus infringe the rules of Petri sport. Such a rule infringement is penalized by a *rollback*, as explained below.

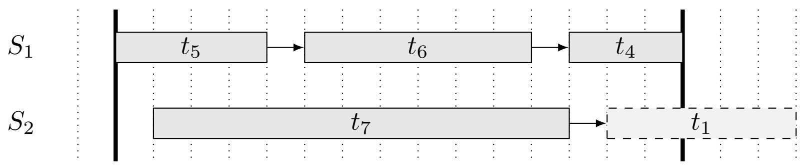


Fig. 5: A transaction consisting of two sequences S_1, S_2 and five transitions t_5, t_6, t_4, t_7 and t_1. The attempt to fire t_1 does not complete before the end of the interval and thereby triggers a *rollback*. The interval limits (beeps) are represented by the thick lines.

Rollback Transition firings have to finish before the end of the current clock interval. If any transition firing does not finish before the next beep, the entire transaction has to be reset and no points are awarded for the entire transaction. In Figure 5 this means that a firing of t_1 will result in all transitions of both sequences to be rolled back, and the marking at the beginning of the transaction is established.

Thus, this rule prevents "opportunistic" transition firings with low chance of success, but also adds a suspense factor for "close transactions". This behavior forces the players to assess the risk of firing several transitions versus the safety of firing only one. It also gives more value to parallel firings, that are difficult to coordinate.

Sudden death Additionally to the rollback an additional *sudden death* penalty is introduced. Sudden death dictates that if a transaction has been executed in a faulty manner (e.g. not enough/too many players are involved in a transition), the transition is reset and the transaction stops, but additionally one free player is excluded from the game round and can no longer participate during the whole round. The whole transaction is not rolled back, as there is already a penalty for the fault, but all other transitions that were being fired in parallel are canceled.

2.6 Winning conditions

A round of Petri sport ends if either of the following conditions are met:

Timeout The game time is over and a predefined number of beeps is reached.

Deadlock No further transitions are enabled, naturally (being trapped due to game design), or due to sudden death (i.e. loss of free players making it impossible to satisfy postconditions).

Livelock While it is possible to fire transitions, it is not possible to reach a state where the team can still score points i.e. only (cycles of) non-point-scoring transitions are enabled.

In either case, the game ends, and the team's score (i.e. the total number of transitions that were fired) is announced. In a competitive settings, the second condition does not trigger the end of the game for the other team(s), but excludes the one that reached a deadlock from remaining rounds. It is then only at the end of the game time that the team with the highest number of fired transitions is declared victorious.

2.7 Refereeing

In most games and sports there are situations where even the fairest players may not reach consensus. In Petri sport, disputes over point counts or proper completion of a transition firing can arise easily – in particular near the end of a game. One or several impartial referees shall be appointed to settle these disputes. Such referee(s) are also responsible for sanctioning illegal moves, usually resulting in a rollback or sudden death. In case several referees are appointed, only one shall have the right to dispense additional penalties, such as point and/or time deductions, or rollback without timer suspension (i.e. time passes as the players run back to last valid marking before the foul play), if deemed necessary. As in any other sport, the ruling of a referee shall not be contested.

3 Formal Definition

Petri sport is played on a special kind of Petri nets, called *sport Petri nets* (SpPN). Their definition has been carefully designed in order to keep the Petri sport field as simple as possible, while still allowing complex behaviors.

3.1 Sport Petri nets

Sport Petri nets are a variant of classical Petri nets. Their most distinguishing feature is that the global number of tokens is bounded. At the beginning of the game a finite number of tokens is split into bound tokens (the initial marking) and free tokens. When a transition is fired, bound tokens become free and free tokens are bound according to preconditions and postconditions. The total number (free + bound) of tokens within the system cannot increase. Every SpPN can be converted into an equivalent place-transition net by adding a place for free tokens and additional pre- and postconditions for each transition, as shown in Figure 3. The following section will elaborate on the formal aspect of sport Petri nets.

We denote S as the set of all SpPN and define that each sport Petri net $s \in S$ is composed of $s = \langle P, T, pre, post, f, i \rangle$.

P	set of places	T	set of transitions
$pre : P \times T \to \mathbb{N}$	preconditions	$f : \mathbb{N}$	free tokens
$post : T \times P \to \mathbb{N}$	postconditions	$i : P \to \mathbb{N}$	initial marking

As typical for Petri nets, SpPN make use of two kinds of arcs: *pre* and *post*, defining the pre- and postconditions of transitions. Arcs are labeled with their *weight*, i.e. a natural number defining the amount of tokens taken from a place or put into it when firing a transition. Arcs with $weight = 0$ do not influence a transition and are hence treated as non-existent.

Markings A SpPN's marking defines the number of tokens within each state and the number of free tokens available. It is thus a function from places (and the free tokens counter) to natural numbers: $m : P \cup \{free\} \to \mathbb{N}$. The marking of a particular place is $m(p)$ (for $p \in P$), and the number of free tokens is given by $m(free)$. The set M of all possible markings of an SpPN s is defined by $M =$

$\{m \mid m : P \cup \{\mathit{free}\} \to \mathbb{N}\}$. The initial marking of the SpPN adds the number of free tokens to m_0. It is thus given by $m_0(\mathit{free}) = f \wedge \forall p \in P, m_0(p) = i(p)$.

A few convenience functions render the below definitions more readable: $\mathit{pre}(\circ, t)$ and $\mathit{post}(t, \circ)$ return the marking corresponding to the precondition (respectively postcondition) of transition t, that includes the free tokens. $+$ and $-$ are addition and subtraction of markings. A special case of using the $-$ operator is to remove a token from the set of free tokens: $m - \{\mathit{free} \mapsto 1\}$ performs this operation.

$$\forall t \in T, \mathit{pre}(\circ, t) = \begin{cases} p & \mapsto \mathit{pre}(p, t), \forall p \in P \\ \mathit{free} & \mapsto \sum_{p \in P} \mathit{post}(t, p) \end{cases}$$

$$\forall t \in T, \mathit{post}(t, \circ) = \begin{cases} p & \mapsto \mathit{post}(t, p), \forall p \in P \\ \mathit{free} & \mapsto \sum_{p \in P} \mathit{pre}(p, t) \end{cases}$$

$$\forall p \in P \cup \{\mathit{free}\}, m_l + m_r = m_l(p) + m_r(p)$$

$$\forall p \in P \cup \{\mathit{free}\}, m_l - m_r = m_l(p) - m_r(p)$$

$$m - \{\mathit{free} \mapsto n\} = \begin{cases} p & \mapsto m(p), \forall p \in P \\ \mathit{free} & \mapsto m(\mathit{free}) - n \end{cases}$$

3.2 Operational Semantics

SpPN's semantics are provided in the remainder of this subsection. It is given using inference rules written as follows, where the semantic domain is split into three parts: *input* $\times$ *parameter* $\times$ *output*.

$$\texttt{name:}\ \frac{}{\langle \dots \rangle \in \mathit{input} \xrightarrow{\texttt{rule}(\cdots \in \mathit{parameter})} \langle \dots \rangle \in \mathit{output}}$$

The semantic rules and their domains are:

$$\texttt{transition} : M \times T \times (M \times \{\top, \bot\})$$

$$\texttt{transaction} : M \times T^* \times M$$

All rules take a marking as input and return a marking. The `transition` rule is parameterized by the transition to fire, and returns a marker ($\top$ or $\bot$) of validity of the firing. The `transaction` rule is parameterized by a sequence of transitions to fire, T^* denotes a sequence of T.

`transition-correct` and `transition-incorrect` are the firing rules for individual transitions. They are parameterized by the transition t to fire, take as input a marking (containing also the free tokens), and return the resulting marking together with a marker $(\top, \bot)$ stating if the firing is correct.

$$\texttt{transition-correct:}\frac{pre(\circ,t) \leq m, m' = m - pre(\circ,t) + post(t,\circ)}{\langle m \rangle \xrightarrow{\texttt{transition}(t)} \langle m', \top \rangle}$$

$$\texttt{transition-incorrect:}\frac{pre(\circ,t) > m \vee m' \neq m - pre(\circ,t) + post(t,\circ)}{\langle m \rangle \xrightarrow{\texttt{transition}(t)} \langle m', \bot \rangle}$$

The `transition-incorrect` rule differs from most of the semantics given for Petri nets, because it explicitly states to what constitutes an invalid firing of a transition. There can be thus an infinity of markings m' as output of this rule. They correspond to all the firings of the transition where players do not follow the firing rules of the Petri net, for instance by using less or more tokens than expected, or tokens from or to wrong places. The failure marking m' is not reused in other semantic rules, the only important part of the result being the $\bot$ marker.

`transaction-empty`, `transaction-correct` and `transaction-incorrect` are the rules for transactions (sequences of transitions). A sequence is denoted as $t..s$ where t is the transaction's first transition, and s the rest of the sequence. [] denotes an empty sequence (i.e. no transitions).

The transitions within a transaction are fired in sequence, the marking being updated one by one. When the firing of a transition within a transaction fails (returns $\bot$), the remainder of the transaction is not performed. Instead, one free token is removed, and the state remains in the last valid marking.

$$\texttt{transaction-empty:}\frac{}{\langle m \rangle \xrightarrow{\texttt{transaction}([])} \langle m \rangle}$$

$$\texttt{transaction-correct:}\frac{\langle m \rangle \xrightarrow{\texttt{transition}(t)} \langle m', \top \rangle, \langle m' \rangle \xrightarrow{\texttt{transaction}(s)} \langle m'' \rangle}{\langle m \rangle \xrightarrow{\texttt{transaction}(t..s)} \langle m'' \rangle}$$

$$\texttt{transaction-incorrect:}\frac{\langle m \rangle \xrightarrow{\texttt{transition}(t)} \langle m', \bot \rangle}{\langle m \rangle \xrightarrow{\texttt{transaction}(t..s)} \langle m - \{f \mapsto 1\} \rangle}$$

This semantics is very abstract and close to the one for common Petri nets, except for the handling of erroneous firings. It cannot express the parallel firing of transitions, as it only allows sequences of firings. Additional rules are provided in Section 3.3 to describe the detail of the SpPN evolution and in particular the ability to fire simultaneously transitions.

3.3 Concurrency, sudden-death and failures

When playing Petri sport, several transitions can be fired in parallel during a transaction. To describe the semantics of such firings, we do not provide the usual parallel composition operator ($||$), but instead decompose each transition into two phases: 1. its resource acquisition phase, where the preconditions are met and free tokens are reserved for the postconditions, 2. its resource release phase where the actual firing happens. Parallel transitions are a sequence of resource acquisition and release phases.

The corresponding semantics thus works on traces, that explicitly list the actions performed by the players and the environment. A trace is a sequence of actions, each action can be one of the following:

- $begin(t, m)$ representing the gathering of preconditions of the transition,
- $end(t, m)$ representing the release of postconditions of the transition,
- *beep* for a *beep* from the timer, i.e. the end of one clock interval and start of another.

The semantics consider all possible traces, not only correct ones. I.e. there are no guaranteed *begin*-*end* pairs, or there can be a *beep* before other actions.

The relations and their domains are given as follows. All take at least one marking as input and return at least one marking. The `action` subrule takes as input and output a multiset of transitions, that have begun but not ended. It also uses the action to perform, and returns a marker of success. The `transaction` subrule takes as input the state in which the system is at the beginning of the transaction. It is used for canceling the transaction. It also takes the multiset of begun transitions. The `rollback` subrule is used to perform rollbacks, when a part of a transaction is canceled, mainly because of sudden death. It takes as parameter the multiset of begun transitions to rollback.

$$\begin{aligned} A &= \{begin(t, m) \mid t \in T, m \in M\} \\ &\cup \{end(t, m) \mid t \in T, m \in M\} \\ &\cup \{beep\} \\ \texttt{action} &: (M \times \mathcal{M}(T)) \times A \times (M \times \mathcal{M}(T) \times \{\top, \bot\}) \\ \texttt{transaction} &: (M \times M \times \mathcal{M}(T)) \times A^* \times M_s \\ \texttt{rollback} &: M \times \mathcal{M}(T) \times M \end{aligned}$$

The multiset of values within a set S is written $\mathcal{M}(S)$. Operations defined on sets are extended on multisets. For instance $x \setminus \{y\}$ only removes one occurrence of y in x if x is a multiset.

The following rules define the behavior of individual steps *begin* and *end*. They are parameterized by the action to perform, take as input a marking and the multiset of transitions that have only begun (those that have seen a *begin* but no *end*), and return the updated marking, the updated list of begun transitions, and a success marker.

$$\texttt{pre-correct:}\frac{m' \leq m, m' = pre(\circ, t)}{\langle m, ts\rangle \xrightarrow{\texttt{action}(begin(t,m'))} \langle m - m', ts \cup \{t\}, \top\rangle}$$

$$\texttt{post-correct:}\frac{t \in ts, m' = post(\circ, t)}{\langle m, ts\rangle \xrightarrow{\texttt{action}(end(t,m'))} \langle m + m', ts \setminus \{t\}, \top\rangle}$$

The following rules are similar, but define the behavior in case of incorrect firing. Several problems are handled: the transition to fire can be disabled, the firing can be incorrect because not enough or too many tokens are used, or a transition is ended without having been begun.

$$\texttt{pre-incorrect:}\frac{m' > m \vee m' \neq pre(\circ, t)}{\langle m, ts\rangle \xrightarrow{\texttt{action}(begin(t,m'))} \langle m, ts, \bot\rangle}$$

$$\texttt{post-incorrect:}\frac{t \notin ts \vee m' \neq post(\circ, t)}{\langle m, ts\rangle \xrightarrow{\texttt{action}(end(t,m'))} \langle m, ts, \bot\rangle}$$

When a transaction comes to its end (there are no more actions to perform), it is validated by looking at the transitions that are still not ended. If one such transition remains, there is an incorrect behavior, leading to a sudden death. This rule does not handle the case where a *beep* occurs, but instead the case where players have forgotten to perform the postcondition.

$$\texttt{empty-correct:}\frac{ts = \varnothing}{\langle m_i, m, ts\rangle \xrightarrow{\texttt{transaction}([])} \langle m\rangle}$$

$$\texttt{empty-incorrect:}\frac{ts \neq \varnothing, \langle m\rangle \xrightarrow{\texttt{rollback}(ts)} m'}{\langle m_i, m, ts\rangle \xrightarrow{\texttt{transaction}([])} \langle m' - \{f_s \mapsto 1\}\rangle}$$

The firing of a transaction performs actions one by one. If one action fails, the marking is restored to its value before this transition, all currently begun transitions are rolled back, and a sudden death occurs *after* the rollback. This information about when the sudden death occurs is important as the free players may change between the rollback. Note that in these semantics the sequence consists of *actions* rather than *transitions* (as in the one above).

$$\texttt{transaction-correct:}\frac{\langle m, ts\rangle \xrightarrow{\texttt{action}(x)} \langle m', ts', \top\rangle, \langle m_i, m', ts'\rangle \xrightarrow{\texttt{transaction}(s)} \langle m''\rangle}{\langle m_i, m, ts\rangle \xrightarrow{\texttt{transaction}(x..s)} \langle m''\rangle}$$

$$\texttt{transaction-incorrect:}\frac{\langle m, ts\rangle \xrightarrow{\texttt{action}(x)} \langle m', ts', \bot\rangle, \langle m\rangle \xrightarrow{\texttt{rollback}(ts)} m''}{\langle m_i, m, ts\rangle \xrightarrow{\texttt{transaction}(x..s)} \langle m'' - \{f_s \mapsto 1\}\rangle}$$

Within a transaction, a `beep` might occur, triggering the beep rule which reverts a marking to the original one.

$$\texttt{beep:}\frac{}{\langle m_i, m, ts\rangle \xrightarrow{\texttt{transaction}(beep..s)} \langle m_i\rangle}$$

`rollback` is used to rollback all transitions that have begun but not ended.

$$\texttt{rollback-empty:}\frac{}{\langle m\rangle \xrightarrow{\texttt{rollback}(\varnothing)} \langle m\rangle}$$

$$\texttt{rollback-nonempty:}\frac{\langle m + pre(\circ, t)\rangle \xrightarrow{\texttt{rollback}(ts)} \langle m'\rangle}{\langle m\rangle \xrightarrow{\texttt{rollback}(\{t\}\cup ts)} \langle m'\rangle}$$

At the beginning of each transaction, the set of `transaction` relation is used with $\langle m_i, m, ts\rangle = \langle m, m, \varnothing\rangle$, where m is the state of the sport Petri net at the beginning of the transaction. When starting the game, this state is m_0.

All transactions that are correct while playing Petri sport are also correct within the sequential semantics and the parallel one. The only difference between the two semantics is that the sequential one transforms parallel firings into sequences. Similarly, all transactions that are incorrect while playing Petri sport are also incorrect within the two semantics (except for *beep* which is not represented in the sequential semantics). Erroneous firings explicitly represented in the parallel semantics are handled by the `transition-incorrect` rule of the sequential semantics.

According to the semantics, the marking graph of a sport Petri net can be computed, but may be infinite when using the simple semantics without concurrency, or when dealing with infinite transactions. In all other cases, assuming that the number of free tokens acts as a bound, the marking graph is guaranteed to be finite. In addition, the bound of the SpPN is lower than the sum of the initially bound tokens and the initially free tokens.

4 Game Variants

There are two ways in which the rules of Petri sport can be modified, so as to suit more experienced players or provide interesting, alternative play styles. The first focuses on the semantics of the underlying Petri net formalism. In the previous sections, we presented a semantics based on classic (*black token*) Petri nets. But Petri sport can very well be adapted to borrow concepts from Petri net extensions, such as colored tokens, inhibitor arcs and similar. Evidently, the rules of a particular flavor of Petri net that would include these extensions must be changed accordingly. For instance, one could imagine assigning colors to players, and designing postconditions so that the number and/or color of the free players that could satisfy them would depend on the number and/or color of the players who satisfied the precondition.

For spatial reasons, we cannot delve deeper into that direction, and therefore will not provide a precise semantical specification of these concepts. Instead, the remainder of this section will focus on a second way of modification: the rules of the game itself. In particular, we will show how minor modifications can create interesting variants to accommodate different play styles.

4.1 No sudden death

As described above, the sudden death rule changes Petri sport's semantics with regards to the Petri net formalism. We noticed that novice players still unfamiliar with the formalism, often find it difficult to adapt their strategy in function of a variable number of free players. Our experiences have shown that it may sometimes be advantageous to remove the sudden death rule, replacing it with other kind of penalties, such as deductions to the available time or awarding of negative points. This allows players to become familiar with the formalism, before having to process more elaborate considerations, such as a potential loss of available players.

4.2 Exponential points

According to the rules presented above, teams are rewarded if they successfully fire several transitions in a given time span. Unfortunately, in complex SpPN designs which require elaborate team coordination, teams might be tempted to play safely and avoid firing long sequences. Indeed, the risk of losing free players under sudden death might discourage teams to try more difficult moves.

This problem can sometimes be overcome without resorting to a complete game redesign. By simply awarding a number of points exponential to the number of fired transitions during a single clock phase the reward for risk becomes higher. For instance, one could imagine awarding a single point for the first fired transition, but then multiplying this number for each subsequent transition (e.g. two points for the next one, four points for the following, etc.) Such a rewarding scheme encourages more risky behavior and creates games with a higher "gambling" character.

4.3 Multi-point transitions

In a usual game, the number of points grows quickly. A standard 5-minute game with a 15 second base clock provides 20 intervals. Considering that in each interval multiple transitions may be fired in parallel and/or in sequence, it is easy to see how scores can quickly reach high numbers. By itself, this is not necessarily bad game design, but can make it harder for teams to properly think out their strategy. Moreover, counting fired transition can prove quite difficult with respect to refereeing, in particular remembering that transitions may be fired in parallel.

So as to ease these aspects, one possible adjustment is to assign a particular value to each transition, or a subset thereof. For instance, this could mean awarding points only if one or several specific transitions are fired, while firing other transitions would not impact the score. An indirect advantage of this variant is that it enables the design of play fields that feature a complex sequence of transition, requiring higher efforts (e.g. long running distances), but whose firing is extremely rewarding. This can also encourage teams to find the quickest ways to repeatedly reach valued transitions.

We also introduce multi- and high-value transitions. By firing these special transitions teams are rewarded with a point multiplier (i.e. doubles the amount of points scored in the current clock interval). Typically such transitions require an elaborate team coordination. Game designers are encouraged to make it hard to enable such transitions and/or repeatedly execute them.

4.4 Colors

While the variants above represent small modifications of the game, we also looked into teaching more complex flavors of Petri nets. In particular, we extended Petri sports to include colors. In color Petri sports, tokens and free players have access to colored shirts/bibs. Wearing a shirt identifies the player as token of a certain color. Each bound player is associated with one specific color which cannot change until freed. Free tokens are color-less and can choose to become any one of the game colors when they participate in a transition. In each game round the game organizers can decide for each color whether it is bounded, i.e. there is maximum number of tokens in a certain color (and hence shirts/bibs), or unbounded (there should be as many bibs of that color as number of players. This game variant significantly changes the original Petri sport semantics. For spatial reasons we cannot provide the semantics of this modification.

4.5 Puzzle mode

Puzzle mode is a modification of Petri sport that reduces the physical aspect of the game. We consider games of this style to be a kind of intellectual exercise. In puzzle mode, the field provides a difficult problem and requires players to find a sequence of transitions that will allow the reaching of a certain goal. This goal can be the enabling and firing of a certain transition, or the reaching of a predefined marking. The teams participating will find themselves directly competing against each other. This means that a before unknown playing field and/or winning condition will be presented to all teams at the same time. The teams can attempt to solve the puzzle on a first-come-first-served basis. The first team to find the solution wins the round. If a team attempts and fails, it is not allowed to try again in the same round. A positive trait of this game variant is that it can also be played by a single team individually. In this case the goal can be to check if the reaching of the goal is possible in the first place, rather than finding the solution fast.

5 Quality metrics

In order to assert the quality of Petri sport we developed guidelines to help the design of sport Petri nets. Below, we elaborate on some quality aspects of SpPN. The suggestions however do not guarantee good SpPN design. It is possible to create an entertaining SpPN that violates some of these rules or create an adhering SpPN that is not playable.

Formally expressed, a SpPN should allow users to fire transitions without reaching deadlock situations. This means that there should be a sequence of transitions (a *path*) whose execution takes longer than the predefined playing time. It has to be taken into account that transitions are only fired between *beeps*, but that it is possible to fire multiple transitions at once. It should further be considered that the duration of a transition firing depends on its complexity, i.e. how many players from how many places are involved, and the physical length of the arcs. Usually the firing of two to three transitions within one beep-interval supports entertaining games.

For the overall entertainment we expect that variety is asserted by making it possible to reach many different game states (markings). In practical terms we recommend that in most game states there should be at least two transitions enabled that lead to different new markings.

Obviously, the presence of deadlocks increases the excitement potential. They should however be sparsely used, in order to keep a good game *flow* [HSR+16].

In terms of team size, game organizers should keep in mind that the sudden death penalties might easily remove some players for the current game round. This means that team sizes should be chosen so that the playability is kept up even during the loss of players.

Playability We define that playability is the set of metrics that give information about the suitability of a SpPN for Petri sport. Examples of unsuitable SpPN include nets that do not have enough free tokens to fire any transition, or do not contain cycles in their state space. These measures are mostly related to standard Petri net key figures such as path length, size of state space, etc.

Entertainment potential Entertainment potential is a measure of how interesting it should be to play a SpPN. In order to standardize this rather subjective figure, we try to provide objective characteristics for evaluation. We propose the following metrics to assess the entertainment potential of a SpPN. All properties are decidable on a finite marking graph.

Number of states A SpPN with only a few reachable states is not interesting, because players will repeat the same actions over and over.

How many deadlock states? A SpPN should contain the risk of being trapped in a deadlock in order to keep the players focused on the game.

Number of states leading only to deadlocks States leading only to deadlocks are not of interest since players cannot escape. The number of such states should thus remain low, compared to the total number of states.

Number of enabled transitions per state Having more than two enabled transitions allows the players to choose which one to fire. They have to make a decision and coordinate, which makes the game more fun.

Number of parallel transitions per state Firing several transitions in parallel is a risk in the SpPN, because it allows players to gain more points, but at the increased risk of sudden death, because coordination is more difficult. Thus, allowing parallel transitions in most of the states is a good practice.

The `petrisport` tool[2] is provided to game organizers to help the analysis of the game and hence provide indicators for playability and entertainment poten-

[2] `https://github.com/cui-unige/petrisport`

tial. The tool calculates these characteristics based on a sport Petri net provided in the PNML format.

Evaluation of the example Figure 6 shows some characteristics of the SpPN in Figure 4 as computed by the `petrisport` tool. The histograms show the ratio between the number of states that have a property and the total number of states in the marking graph. In total three characteristics are displayed as histograms: 1. *choice ratio* expresses the fraction of system states that offer at least two enabled transitions, 2. *parallel ratio* considers the states that can fire at least two transitions in parallel, 3. *deadlock ratio* shows the states that can only lead to deadlocks. Line plots show the mean number of *enabled transitions* and *parallel firings* in each state.

The analysis of the SpPN of the example shows that it can be fun starting from 6 free tokens (thus 8 players in total). As sudden death may occur during the game, we recommend to start the game with 10 or 12 players, in order to avoid reaching the less entertaining token numbers too soon.

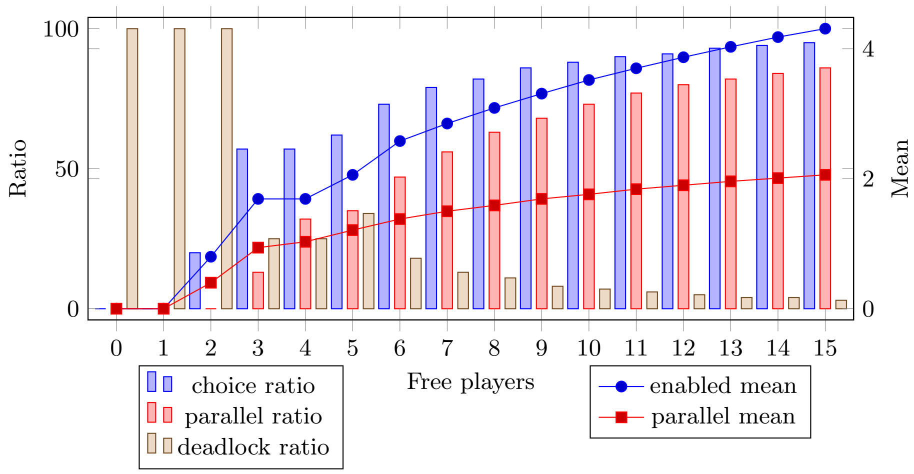


Fig. 6: Characteristics of the sport Petri net of Figure 4 by number of free tokens in the initial marking.

6 Experimental Evaluation

Petri sport has been successfully played at the University of Geneva. In this section we present initial experiment results and participant reactions after playing. Note, that these results are preliminary: the feedback has been captured using a simple, anonymous questionnaire and represents the participants' opinions of the game. So far, there was no controlled experiment that formally attempts to measure the impact Petri sports has on students' knowledge or academic success.

Methodology We played the game with undergraduate (Bachelor's degree) and graduate (Master's degree) students who learned about, and used Petri nets in previous academic courses before playing the game.[3]

Participants were taught the rules of the game gradually, by introducing the different notions one by one. The introduction was played on the "introductory" SpPN (Figure 7), which while simple exposes all interesting concepts, such as source and sink transitions, split, merge and synchronization. Each notion was introduced by a short theoretical information followed by some practice. We proceeded in the following steps:

1. We introduced the basic concepts (i.e. movements along the arcs, firing transitions, consumption and production of tokens).
2. We introduced the synchronization mechanisms for transitions involving multiple preconditions.
3. We invited the students to do trial play without timing.
4. We introduced the clock-based system (i.e. transition have to finish before interval end).
5. We introduced the penalty system (i.e. the reset rule and sudden death rules).
6. We invited the students to a practice round with all rules in full-force.
7. We carried out a competitive round.

Since the team sizes were held minimal to allow multiple competitive teams, the experiments were played using the *no-sudden-death* variant, where faulty firing leads to a reset of the transition.

After an initial round on the "introductory" SpPN, the participants were given a picture of a second, more complex SpPN (depicted in Figure 4, above). Following a few minutes of study, and one minute of setup, the teams were left to play the game in two rounds. In the former, students were left to play without additional instructions, while in the latter they were encouraged to target the outer transition with the longer running paths and try to score higher than before. The complete session lasted about 90 minutes.

Evaluation Following the experiment, the students answered a short survey where they answered questions using a five-valued scale from 0 (disagree) to 5 (agree) and open answers. While a full discussion would exceed the scope of this publication, we will summarize the most important feedback.

Most students enjoyed playing Petri sports, independent of their familiarity with the formalism. Additionally, despite having already completed at least one course, almost all students responded to having learned "a bit" more about Petri nets while playing the game. About half of the students felt that playing Petri sports would have strongly helped them understand Petri nets when they first came in contact with the formalism. When asked whether Petri sports should be included into the undergraduate course that first introduces Petri nets, students provided mixed answers. About half of the participants were interested in using the game as a means to learn about Petri nets.

[3] We have yet to experiment with complete novices.

Discussion The results we obtained encourage us to continue using Petri sports as a means of teaching. Petri sports positions itself as a fun addition to classical lectures and allows us the playful introduction of Petri net concepts. We do see however, that the students' learning types and opinions on alternative teaching methods differ largely and traditional classroom methods cannot be replaced.

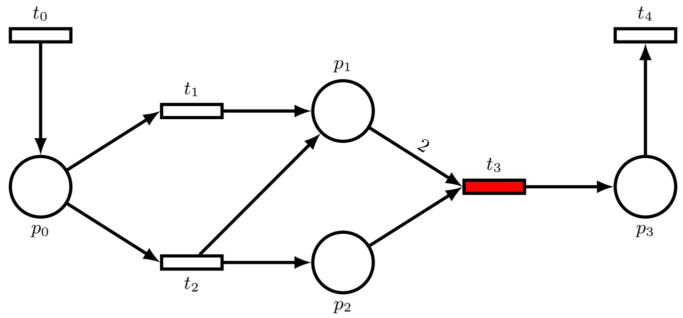


Fig. 7: Introductory sport Petri net that can be used as a tutorial. Teams of at least five players can learn about source/sink transitions, synchronization, splitting and merging.

7 Related Work

The use of games for education purposes is a hot topic in the didactics domain [EGCN+11,SPS05]. In particular, approaches based on games met remarkable success in the context of teaching activities, in the form of classroom games. Also interesting to note is the surge of toys, games and other applications that aim at teaching algorithmic thinking – or similar concepts – to young students [Sqi05,RMMH+09,Boe17]. The use of (digital) games for other purposes than pure entertainment (e.g. education, training, activism, health awareness, etc.) is referred to as *serious gaming* [MC05]. This relatively new domain explores how to best design games so that the learning experience is as natural and efficient as possible, while maximizing the entertainment aspect to make solutions attractive to young audiences [EN11].

The omnipresence of personal computers, gaming consoles and smartphones also pushes people of all demographic groups to disrupt traditional learning approaches. The number of mobile applications and online platforms seeking to teach about subjects like (spoken) languages, geography or biology is growing by the day. Despite the reluctance of some professionals at first, their relevance has became undeniable [VG12].

For nearly a decade now, our group has embraced such untraditional approaches for its teaching activities, and in the context of campaigns promoting university studies. We conducted projects with LEGO robots to introduce formal verification concepts [LRC$^+$14], illustrated formalisms and algorithms with more common and familiar concepts such as cooking [LR16], and used 3D printed devices to concretize otherwise abstract notions such as memory or communication [RB18].

8 Conclusion and Future Works

This paper presents Petri sports, a novel approach to teaching the Petri net formalism to domain novices. Petri sports is motivated by the experience that classroom Petri net teaching often leads to misunderstandings and an increased need for practice and exercise with the formalism. Through gamification, we created Petri sports which allows students to "act as" Petri net tokens and thereby access an additional learning method. Petri sports uses the standard Petri net syntax to design playing fields and its semantics to prescribe movement of players across the pitch. Additional rules bound the number of tokens, and ensure intellectual challenges through *sudden death* and *rollback* penalty rules. All Petri sports rules are based on the formalization provided and allow for programmatic analysis of game characteristics (such as detection of deadlock reachability, etc.) as shown in this paper. Petri sports is extensible and adaptable to best serve the participants' knowledge, physical abilities and preferences. The paper presents several game variants to classical Petri sports and briefly introduces colored Petri sports, puzzle mode and other variations to the game.

The experimental results presented in this paper show that , our While our basic work shows good results in the classroom, we plan to extend Petri sports in several ways:

1. We use numerous game variants to explain various advanced Petri net concepts (inhibitor arcs, colored Petri nets, etc.). We are in the process of creating a catalog thereof, categorically documenting specific rules, semantics and educational success.
2. As we observe good results, we would like to spread Petri sports to other institutions and learn from their experiences as well.
3. Petri sports shows the success of gamification concepts in Petri net teaching. We are currently evaluating whether other gaming styles (such as mobile or browser games) can be used to teach Petri nets. An early stage mobile application project can be found on our group's blog[4].
4. In parallel to the above extension of Petri net teaching, we also evaluate whether other formal methods domains can be taught more easily using such alternative forms of teaching.
5. We are planning to perform evaluations together with experts in the didactic domain, in order to evaluate the teaching success of Petri sports more formally.

[4] `https://cui-unige.github.io/blog-smv/2018/03/26/petri-ios/`

References

Boe17. Benjamin Boettner. Harvards wyss institute launches root robotics, an educational startup to get anyone coding, Aug 2017.

EGCN+11. Alejandro Echeverria, Cristian Garcia-Campo, Miguel Nussbaum, Francisca Gil, Marco Villalta, Matias Améstica, and Sebastian Echeverria. A framework for the design and integration of collaborative classroom games. *Computers & Education*, 57(1):1127 – 1136, 2011.

EN11. Simon Egenfeldt-Nielsen. What makes a good learning game?: Going beyond edutainment. *eLearn*, 2011(2), February 2011.

HSR+16. Juho Hamari, David J Shernoff, Elizabeth Rowe, Brianno Coller, Jodi Asbell-Clarke, and Teon Edwards. Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. *Computers in Human Behavior*, 54:170–179, 2016.

Jen14. K. Jensen. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*. Number v. 1 in Monographs in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, 2014.

Kap12. Karl M Kapp. *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons, 2012.

LR16. David P.Y. Lawrence and Dimitri Racordon. Cooking with algorithms. `https://www.youtube.com/watch?v=aDtq_bLXKvI`, 2016.

LRC+14. David Lawrence, Dimitri Racordon, Maximilien Colange, Steve Hostettler, Alban Linard, Edmundo Lepez Bobeda, Alexis Marechal, Matteo Risoldi, Nicolas Sedlmajer, and Didier Buchs. Introducing formal verication with lego. In *2nd workshop on Fun With Formal Methods, Vienna, Austria, July 13 , 2014, Proceedings*, 2014.

MC05. David R. Michael and Sandra L. Chen. *Serious Games: Games That Educate, Train, and Inform*. Muska & Lipman/Premier-Trade, 2005.

Pet62. Carl Adam Petri. *Kommunikation mit Automaten*. PhD thesis, Universität Hamburg, 1962.

RB18. Dimitri Racordon and Didier Buchs. Démystifier les concepts informatiques par l'éxpérimentation. In *Didapro 7 – DidaSTIC, De 0 à 1 ou l'heure de l'informatique à l'école, Lausanne, Switzerland, February 7-9 , 2018, Proceedings*, 2018.

RMMH+09. Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. Scratch: Programming for all. *Commun. ACM*, 52(11):60–67, November 2009.

SPS05. Julia Shaftel, Lisa Pass, and Shawn Schnabel. Math games for adolescents. *TEACHING Exceptional Children*, 37(3):25–30, 2005.

Sqi05. Kurt Sqire. Changing the game: What happens when video games enter the classroom? *Innovate: Journal of online education*, 1(6):5, 2005.

VG12. Roumen Vesselinov and John Grego. Duolingo effectiveness study. *City University of New York, USA*, 2012.

Zai14. D. A. Zaitsev. Toward the minimal universal petri net. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(1):47–58, Jan 2014.