Test purpose selection using Hit-or-Jump (HoJ) exploration heuristic

In the **test case generation** process (see the model specification tutorial), the objective is to compute a **symbolic subtree** of the reference timed symbolic automaton restricted by a **test purpose**, defined as a **consecutive sequence of transitions** to be covered.

To enable the **selection of such sequences** from the model, the **Hit-or-Jump (HoJ)** exploration heuristic — provided as a dedicated function of **SPTG**, inherited from **Diversity** — can be used. This heuristic guides symbolic exploration toward specific behavioral goals while avoiding exhaustive exploration of irrelevant paths.

The main idea of HoJ is to start from a **declared set or sequence of automata constructs**, which may include:

- Transitions
- States
- Input/output actions or ports

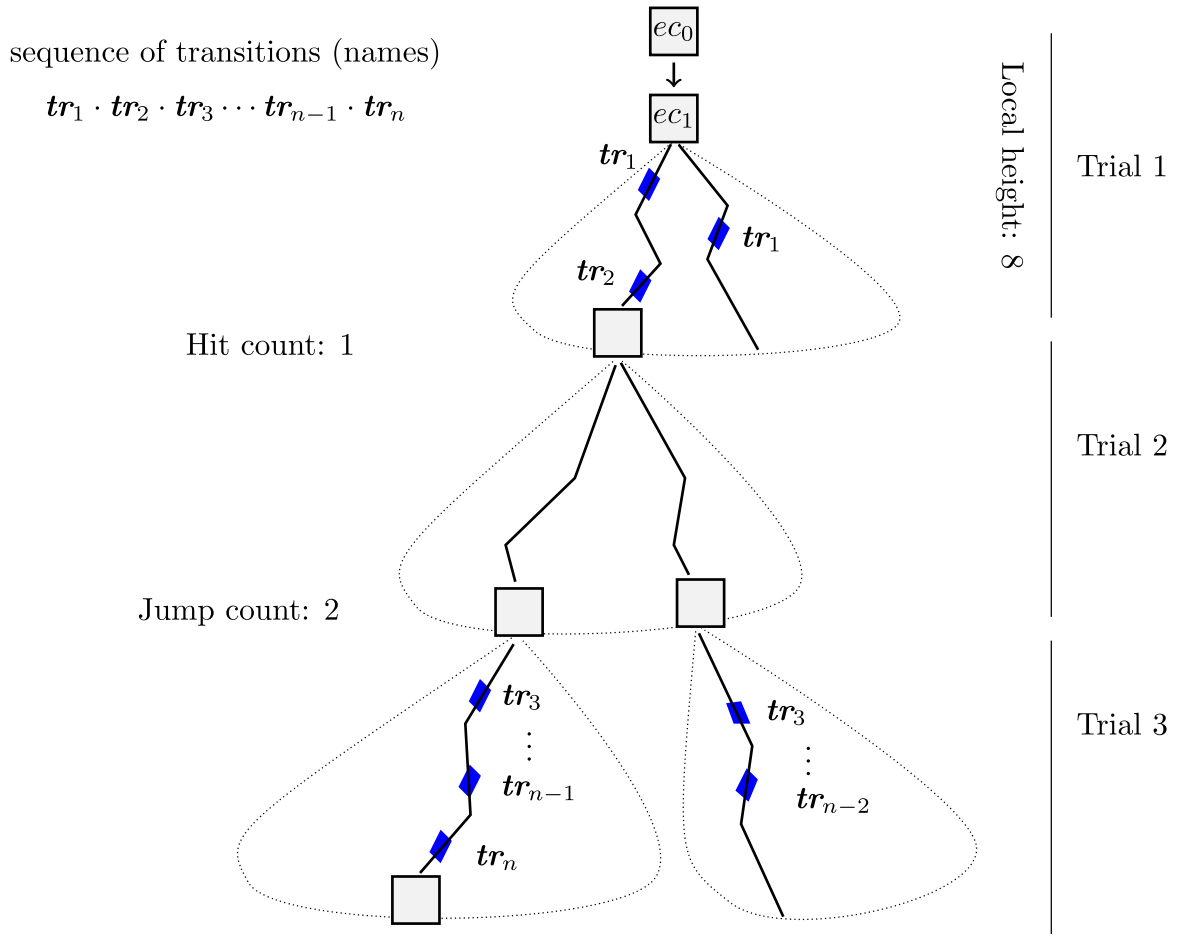
and to iteratively explore the symbolic tree to **find a symbolic path** that satisfies the desired coverage goal.

Once such a path is identified:

- It corresponds to a **consecutive sequence of transitions** in the automaton.
- This sequence serves as the **test purpose** for the **SPTG test case generation** process.

1. Principle of the Hit-or-Jump heuristic

Assuming the **coverage goal** is a **sequence of transitions** which must be covered in order. During symbolic execution, HoJ drives the exploration of the symbolic automaton so that the generated symbolic tree **progressively covers prefixes** of this sequence until full coverage is achieved as illustrated in th the following (Schematic illustration of HoJ trials and coverage progression):



The symbolic tree is built **incrementally and adaptively** through a series of **trials**. In each trial, HoJ computes a symbolic subtree of **bounded local height (N)** using a **breadth-first traversal**. Once the subtree is constructed, it is analyzed to measure the degree of coverage achieved for the target sequence.

- **Hit:** If at least one non-empty prefix of the sequence is covered, HoJ selects, at random, one or several **execution contexts (ECs)** corresponding to the **maximum prefix coverage** and restarts exploration from these ECs.
- **Jump:** If no prefix is covered, HoJ randomly selects one or several ECs in the subtree to restart the breadth-first exploration from their corresponding states.

This process is **iteratively repeated** (Trial 1, Trial 2, ...) until the full target sequence is covered. Each local subtree (bounded by dashed areas) corresponds to one trial, alternating between **Hit** and **Jump** phases until complete coverage is achieved.

2. Coverage Modes

HoJ supports different **coverage modes**, depending on the structure of the declared test purpose and the desired level of strictness:

- **Sequence coverage:** requires transitions to be covered **in the declared order**.
- **Consecutive coverage:** requires that at least **one new element** of the sequence is covered at each iteration.

- **Folding coverage:** allows covering **multiple elements** at the same step (e.g., if several transitions correspond to equivalent or simultaneous actions).

These modes provide flexibility in defining **how tightly the heuristic should follow the declared sequence**, depending on the abstraction level of the model or the granularity of coverage desired.

3. Heuristic Parameters

The HoJ heuristic is controlled by several key parameters that determine its exploration behavior:

- **Local height (N):** the maximal depth of each symbolic subtree computed using BFS during a trial.
- **Hit count:** the number of ECs with maximal coverage selected at random to restart the next BFS exploration in case of a Hit.
- **Jump count:** the number of ECs chosen at random to restart exploration when no coverage progress is observed.
- **Trial count:** the number of allowed re-starts (iterations) of the HoJ process.

Tuning these parameters allows balancing **exploration depth** and **search focus**, ensuring that the heuristic converges efficiently toward a subtree that satisfies the **test purpose**.