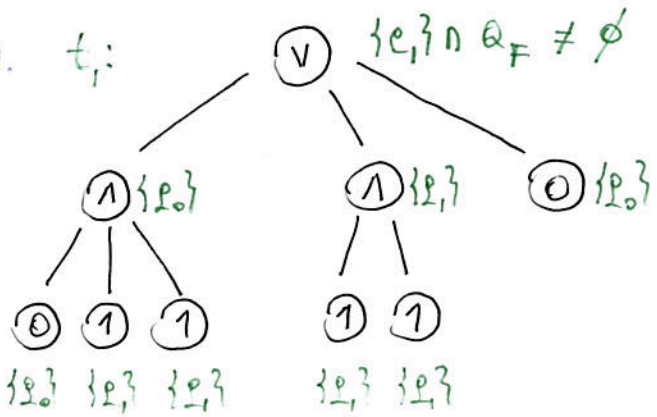
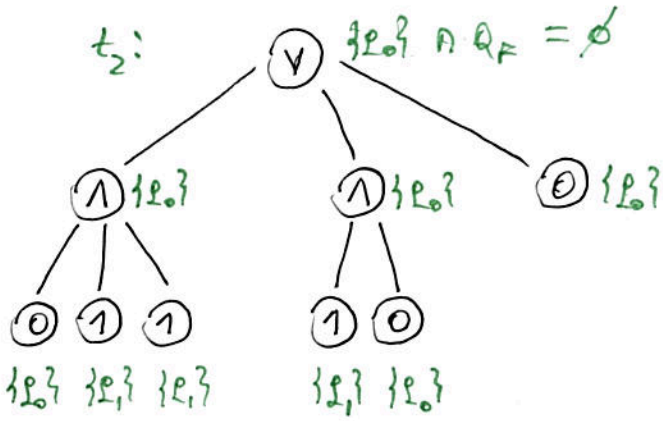
13.1. t_1: 

t_2: 

13.2. (a) A possible XML document answering the task may look as follows:

```
<collection>
<book> —▷   <author>
              <name> Antoine de Saint Exupéry </name>
              <year> 1900 </year>
              <city> Lyon, France </city>
            </author>
            <title> The little prince </title>
            <publisher>
              <name> Houghton Mifflin Harcourt </name>
              <city> Boston, USA </city>
            </publisher>
            <isbn> 0152465030 </isbn>
            <year> 1943 </year>
            <series> The collector's colour library </series>
</book> —▷
            - - -
</collection>
```

(b)
```
<! DOCTYPE COLLECTION [
    <! ELEMENT collection  ( book+ ) >
    <! ELEMENT book        ( author+, title, publisher, isbn, year, series? ) >
    <! ELEMENT author      ( name, year, city ) >
    <! ELEMENT title       ( #PCDATA ) >
    <! ELEMENT publisher   ( name, city ) >
```

```
<!ELEMENT isbn    (#PCDATA)>
<!ELEMENT year    (#PCDATA)>
<!ELEMENT series  (#PCDATA)>
<!ELEMENT name    (#PCDATA)>
<!ELEMENT city    (#PCDATA)>
]>
```

In order to validate the XML document from (a) against the DTD from (b) one must represent the XML document as a tree and check that it belongs to the NHA language described by the DTD.

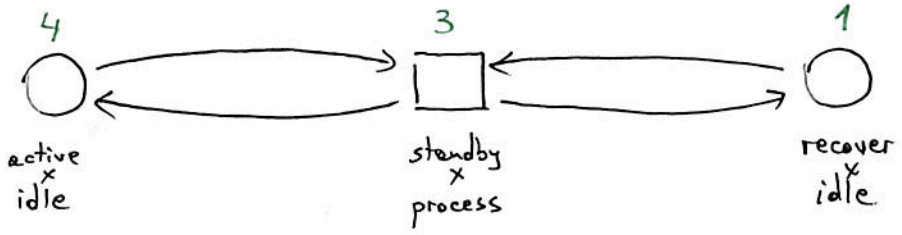
13.3. Let $p^0 \to p^1 \to \dots$ be a play conform to strategy s, as defined in the proof.

Due to wellfoundedness of $p_0 < p_1 < \dots$ one can pick the first p^j of the play that has minimal index wrt. $p_0, p_1, \dots$.

Then $p^j \to p^{j+1} \to \dots$ follows only the strategy $s_{p_{min}}$, so it is a winning suffix of the initial play.

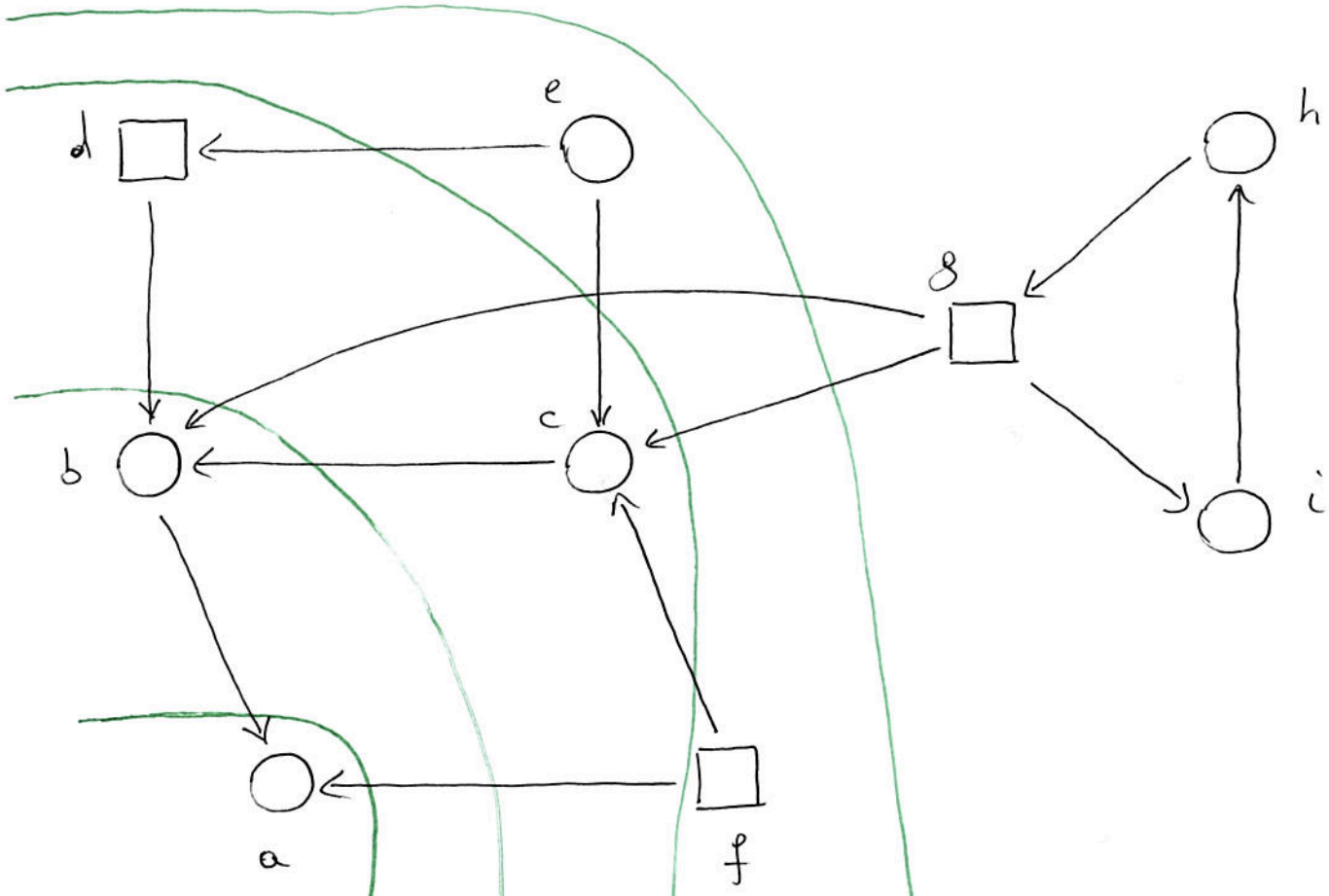
This, however, makes the entire play winning so s is a winning strategy for $\mathcal{U}$.

13.4. (a) The general procedure for constructing a parity game out of the given automata would build their synchronizing cross product. Restricting the result to its reachable part gives



A possible $\mathcal{R}$ labelling of the positions which guarantees the game is won by the client if it can reach its active state infinitely often is displayed on the picture above.

(b) Computing the attractor of $\{a\}$ for A is graphically displayed on the picture below:



$Attr_0^A(a)$ $Attr_1^A(a)$ $Attr_2^A(a)$ $Attr_3^A(a) = Attr^A(\{a\})$

$= \{a, b, c, d, e, f\}$