Expressing Properties of Resource-Bounded Systems: The Logics RTL* and RTL

Nils Bulling[1] and Berndt Farwer[2]

[1] Department of Informatics, Clausthal University of Technology, Germany
[2] School of Engineering and Computing Sciences, Durham University, UK

Abstract. Computation systems and logics for modelling such systems have been studied to a great extent in the past decades. This paper introduces resources into the models of systems and discusses the *Resource-Bounded Tree Logics* **RTL** and **RTL***, based on the well-known *Computation Tree Logics* **CTL** and **CTL***, for reasoning about computations of such systems. We present initial results on the complexity/decidability of model checking.

1 Introduction

The basic idea of rational agents being autonomous entities perceiving changes in their environment and acting according to a set of rules or plans in the pursuit of goals does not take resources into account. However, many actions that an agent would execute in order to achieve a goal can – in real life – only be carried out in the presence of certain resources. Without sufficient resources some actions are not available, leading to plan failure. The analysis of agents and (multi-agent) systems with resources is still in its infancy and has been tackled almost exclusively in a pragmatic and experimental way. This paper takes first steps in modelling resource bounded systems (which can be considered as the single-agent case of the scenario just described). Well-known computational models are combined with a notion of resource to enable a more systematic and rigorous specification and analysis of such systems. The main motivation of this paper is to propose a fundamental formal setting. In the future we plan to focus on a more practical aspect, i.e., how this setting can be used for the verification of systems.

The proposed logic builds on *Computation Tree Logic* [6]. Essentially, the existential path quantifier $\mathsf{E}\varphi$ (there is a computation that satisfies φ) is replaced by $\langle\rho\rangle\gamma$ where ρ represents a set of available resources. The intuitive reading of the formula is that there is a computation *feasible with the given resources* ρ that satisfies γ.

Finally, we turn to the decidability of model checking the proposed logics. We show that **RTL** (*Resource-Bounded Tree Logic*), the less expressive version, has a decidable model checking problem as well as restricted variants of the full logic **RTL*** and its models.

The remainder of the paper is structured as follows. In Section 2 we recall the computation tree logic **CTL*** and define multisets used as a representation for

J. Dix, M. Fisher, and P. Novák (Eds.): CLIMA X, LNAI 6214, pp. 22–45, 2010.

resources. Section 3 forms the main part of the paper. We introduce resources into the computation tree logics and their models. Subsequently, in Section 4 we show some properties of the logics. Section 5 includes the analysis of the model checking complexity, and finally, we conclude with an outlook on future work in Section 6.

2 Preliminaries

In this section we present the computation tree logics **CTL** and **CTL*** as well as multisets which we will use to represent resources.

2.1 Computation Tree Logic and Transition Systems

A *Kripke frame* $\mathcal{T} = (Q, \rightarrow)$ consists of a finite set of states Q and a (serial) binary relation $\rightarrow \subseteq Q \times Q$ between states. We say that a state q' is *reachable* from a state q if $q \rightarrow q'$. A *Kripke model* is defined as $\mathfrak{M} = (Q, \rightarrow, \mathcal{P}rops, \pi)$ where $(Q, \rightarrow)$ is a transition system, $\mathcal{P}rops$ a non-empty set of *propositions*, and $\pi : Q \rightarrow \mathcal{P}(\mathcal{P}rops)$ a *labelling function* that indicates which propositions are true in a given state. Such models represent the temporal behaviour of systems. There are no restrictions on the number of times a transition is used.

A *path* λ of a transition system is an infinite sequence $q_0 q_1 \cdots \in Q^\omega$ of states such that $q_i \rightarrow q_{i+1}$ for all $i = 0, 1, 2, \ldots$. Given a path λ we use $\lambda[i]$ and $\lambda[i, j]$ to refer to state q_i and to the path $q_i q_{i+1} \ldots q_j$ where $j = \infty$ is permitted, respectively. A path starting in q is called *q-path*. The set of all paths in $\mathfrak{M}$ is denoted by $\Lambda_\mathfrak{M}$ and the set of all q-paths by $\Lambda_\mathfrak{M}(q)$.

Formulae of **CTL*** [8] are defined by the following grammar:

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{E}\gamma \quad \text{where} \quad \gamma ::= \varphi \mid \neg\gamma \mid \gamma \wedge \gamma \mid \varphi \,\mathcal{U}\, \varphi \mid \bigcirc\varphi$$

and $\mathsf{p} \in \mathcal{P}rops$. Formulae φ (resp. γ) are called *state* (resp. *path*) formulae. There are two temporal operators: $\bigcirc$ (in the next moment in time) and $\mathcal{U}$ (until). The temporal operators $\Diamond$ (sometime in the future) and $\Box$ (always in the future) can be defined as abbreviations.

CTL* formulae are interpreted over Kripke structures; truth is given by the satisfaction relation in the usual way: For state formulae we have

$\mathfrak{M}, q \models \mathsf{p}$ iff $\lambda[0] \in \pi(\mathsf{p})$ and $\mathsf{p} \in \mathcal{P}rops$;
$\mathfrak{M}, q \models \neg\varphi$ iff $\mathfrak{M}, q \not\models \varphi$;
$\mathfrak{M}, q \models \varphi \wedge \psi$ iff $\mathfrak{M}, q \models \varphi$ and $\mathfrak{M}, q \models \psi$;
$\mathfrak{M}, q \models \mathsf{E}\varphi$ iff there is a path $\lambda \in \Lambda_\mathfrak{M}(q)$ such that $\mathfrak{M}, \lambda \models \varphi$;

and for path formulae

$\mathfrak{M}, \lambda \models \varphi$ iff $\mathfrak{M}, \lambda[0] \models \varphi$;
$\mathfrak{M}, \lambda \models \neg\gamma$ iff $\mathfrak{M}, \lambda \not\models \gamma$;
$\mathfrak{M}, \lambda \models \gamma \wedge \delta$ iff $\mathfrak{M}, \lambda \models \gamma$ and $\mathfrak{M}, \lambda \models \delta$;
$\mathfrak{M}, \lambda \models \bigcirc\gamma$ iff $\lambda[1, \infty], \pi \models \gamma$; and

$\mathfrak{M}, \lambda \models \gamma \,\mathcal{U}\, \delta$ iff there is an $i \in \mathbb{N}_0$ such that $\mathfrak{M}, \lambda[i, \infty] \models \delta$ and $\mathfrak{M}, \lambda[j, \infty] \models \gamma$ for all $0 \leq j < i$;

A less expressive fragment of **CTL*** called **CTL** [6] has become popular due to its *better computational properties.* **CTL** restricts **CTL*** such that every temporal operator must directly be preceded by a path quantifier. The formula $\mathsf{E}\Box \Diamond \mathsf{p}$, for instance, is a formula of the full language but not of the restricted version.

2.2 Multisets

We define some variations of multisets used in the following sections. We assume that $\mathbb{N}_0 = \{0, 1, 2, \dots\}$ and $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$.

Definition 1 ($\mathbb{Z}/\mathbb{Z}^\infty$-multiset,$X^\pm_\infty$, $X^\pm$, $\mathbb{N}_0/\mathbb{N}_0^\infty$-multiset, $X^\oplus_\infty$, $X^\oplus$). *Let X be a non-empty set.*

(a) A $\mathbb{Z}$-multiset $\mathbf{Z} : X \longrightarrow \mathbb{Z}$ *over the set X is a mapping from the elements of X to the integers.*
A $\mathbb{Z}^\infty$-multiset $\mathbf{Z} : X \longrightarrow \mathbb{Z} \cup \{-\infty, \infty\}$ *over the set X is a mapping from the elements of X to the integers extended by $-\infty$ and ∞.*
The set of all $\mathbb{Z}$-multisets *(resp.* $\mathbb{Z}^\infty$-multisets*) over X is denoted by $X^\pm$ (resp. $X^\pm_\infty$).*

(b) An $\mathbb{N}_0$-multiset *(resp.* $\mathbb{N}_0^\infty$-multiset*)* $\mathbf{N}$ *over X is a* $\mathbb{Z}$-multiset *(resp.* $\mathbb{Z}^\infty$-multiset*) over X such that for each $x \in X$ we have $\mathbf{N}(x) \geq 0$. The set of all* $\mathbb{N}_0$-multisets *(resp.* $\mathbb{N}_0^\infty$-multisets*) over X is denoted by $X^\oplus$ (resp. $X^\oplus_\infty$).*

Whenever we speak of a 'multiset' without further specification, the argument is supposed to hold for any variant from Def. 1. In general, we overload the standard set notation and use it also for multisets, i.e., $\subseteq$ denotes multiset inclusion, $\emptyset$ is the empty multiset, etc. We assume a global set of resource types $\mathcal{R}$. The resources of an individual agent form a multiset over this set. $\mathbb{Z}$-multiset operations are straightforward extensions of $\mathbb{N}_0$-multiset operations.

Multisets are frequently written as formal sums, i.e., a multiset $\mathbf{M} : X \longrightarrow \mathbb{N}_0$ is written as $\sum_{x \in X} \mathbf{M}(x)$. Given two multisets $\mathbf{M} : X \longrightarrow \mathbb{N}_0$ and $\mathbf{M}' : X \longrightarrow \mathbb{N}_0$ over the same set X, multiset union is denoted by $+$, and is defied as $(\mathbf{M} + \mathbf{M}')(x) := \mathbf{M}(x) + \mathbf{M}'(x)$ for all $x \in X$. Multiset difference is defined only if $\mathbf{M}$ has at least as many copies of each element as $\mathbf{M}'$. Then, $(\mathbf{M} - \mathbf{M}')(x) := \mathbf{M}(x) - \mathbf{M}'(x)$ for all $x \in X$. For $\mathbb{Z}$-multisets, $+$ is defined exactly as for multisets, but the condition is dropped for multiset difference, since for $\mathbb{Z}$-multisets negative multiplicities are possible. Finally, for $\mathbb{Z}^\infty$-multisets we assume the standard arithmetic rules for $-\infty$ and ∞ (for example, $x + \infty = \infty$, $x - \infty = -\infty$, etc).

We define multisets with a bound on the number of elements of each type.

Definition 2 (Bounded multisets). *Let $k, l \in \mathbb{Z}$. We say that a multiset $\mathbf{M}$ over a set X is* k-bounded *iff $\forall x \in X$ $(\mathbf{M}(x) \leq k)$. We use ${}^k X^\pm_\infty$ to denote the set of all k-bounded $\mathbb{Z}^\infty$-multisets over X; and analogously for the other types of multisets.*

Finally, we define the (positive) restriction of a multiset with respect to another multiset, allowing us to focus on elements with a positive multiplicity.

Definition 3 ((Positive) restriction, $\mathbf{M} \restriction_{\mathbf{N}}$). *Let* $\mathbf{M}$ *be a multiset over* X *and let* $\mathbf{N}$ *be a multiset over* Y*. The* (positive) restriction of $\mathbf{M}$ regarding $\mathbf{N}$, $\mathbf{M} \restriction_{\mathbf{N}}$*, is the multiset* $\mathbf{M} \restriction_{\mathbf{N}}$ *over* $X \cup Y$ *defined as follows:*

$$\mathbf{M} \restriction_{\mathbf{N}} (x) := \begin{cases} \mathbf{M}(x) & \text{if } \mathbf{N}(x) \geq 0 \text{ and } x \in Y \\ 0 & \text{otherwise.} \end{cases}$$

So, the multiset $\mathbf{M} \restriction_{\mathbf{N}}$ equals $\mathbf{M}$ for all elements contained in $\mathbf{N}$ which have a non-negative quantity, and 0 for all others elements.

3 Modelling Resource-Bounded Systems

In this section we introduce *resource-bounded models* (RBMs) for modelling system with limited resources. Then, we propose the logics $\mathsf{RTL}^\star$ and RTL (resource-bounded tree logics), for the verification of such systems. Subsequently, we introduce cover models and graphs and consider several properties and special cases of RBMs.

3.1 Resource-Bounded Systems

A resource-bounded agent has at its disposal a (limited) repository of resources. Performing actions reduces some resources and may produce others; thus, an agent might not always be able to perform all of its available actions. In the single agent case that we consider here this corresponds to the activation or deactivation of transitions.

Definition 4 (Resources $\mathcal{R}$, resource quantity (set), feasible)
An element of the non-empty and finite set $\mathcal{R}$ *is called* resource*. A tuple* $(r, c) \in \mathcal{R} \times \mathbb{Z}^\infty$ *is called* resource quantity *and we refer to* c *as the* quantity *of* r*. A* resource-quantity set *is a* $\mathbb{Z}^\infty$*-multiset* $\rho \in \mathcal{R}^{\pm}_{\infty}$*. Note that* ρ *specifies a resource quantity for each* $r \in \mathcal{R}$*.*

Finally, a resource-quantity set ρ *is called* feasible *iff* $\rho \in \mathcal{R}^{\oplus}_{\infty}$*; that is, if all resources have a non-negative quantity.*

We model resource-bounded systems by an extension of Kripke frames, allowing each transition to *consume* and *produce* resources. We assign pairs $(\mathbf{c}, \mathbf{p})$ of resource-quantity sets to each transition, denoting that a transition labelled $(\mathbf{c}, \mathbf{p})$ *produces* $\mathbf{p}$ and *consumes* $\mathbf{c}$.

Definition 5 (Resource-bounded model). *A* resource-bounded model *(*RBM*) is given by* $\mathfrak{M} = (Q, \rightarrow, \mathcal{P}rops, \pi, \mathcal{R}, t)$ *where*

- Q*,* $\mathcal{R}$*, and* $\mathcal{P}rops$ *are finite sets of states, resources, and propositions, respectively;*

- $(Q, \rightarrow, \mathcal{P}rops, \pi)$ *is a Kripke model; and*
- $t : Q \times Q \rightarrow \mathcal{R}^{\oplus} \times \mathcal{R}^{\oplus}$ *is a (partial)* resource function*, assigning to each transition (i.e., tuple* $(q, q') \in \rightarrow$*) a tuple of feasible resource-quantity sets. Instead of* $t(q, q')$ *we sometimes write* $t_{q,q'}$ *and for* $t_{q,q'} = (\mathbf{c}, \mathbf{p})$ *we use* ${}^{\bullet}t_{q,q'}$ *(resp.* $t^{\bullet}_{q,q'}$*) to refer to* $\mathbf{c}$ *(resp.* $\mathbf{p}$*).*

Hence, in order to make a transition from q to q', where $q \rightarrow q'$, the resources given in ${}^{\bullet}t_{q,q'}$ are *required*; and in turn, $t_{q,q'}{}^{\bullet}$ are *produced* after executing the transition. Note, that we only allow finite productions and consumptions.

A *path* of an RBM is a path of the underlying Kripke structure. We also use the other notions for paths introduced above.

The consumption and production of resources of a path can now be defined in terms of the consumptions and productions of the transitions it comprises. Intuitively, not every path of an RBM is feasible; consider, for instance, a system consisting of a single state q only where $q \rightarrow q$ and $t^{\bullet}_{q,q} = {}^{\bullet}t_{q,q}$. It seems that the transition "comes for free" as it produces the resources it consumes; however, this is not the case. The path $qqq\ldots$ is not feasible as the initial transition is not enabled due to the lack of initial resources. Hence, in order to enable it, at least the resources given in ${}^{\bullet}t_{q,q}$ are necessary. Intuitively, a path is said to be *ρ-feasible* if each transition in the sequence can be executed with the resources available at the time of execution.

Definition 6 (ρ-feasible path, resource-extended path). *A path* $\lambda = q_1 q_2 q_3 \cdots \in \Lambda_{\mathfrak{M}}(q)$ *where* $q = q_1$ *is called* ρ-feasible *if for all* $i \in \mathbb{N}$ *the resource-quantity set*

$$\left(\rho + \Sigma_{j=1}^{i-1}(t^{\bullet}_{q_j q_{j+1}} - {}^{\bullet}t_{q_j q_{j+1}})\right) \restriction_{{}^{\bullet}t_{q_i q_{i+1}}} - {}^{\bullet}t_{q_i q_{i+1}} \text{ is feasible.}$$

A resource-extended *path is given by* $\lambda \in (Q \times \mathcal{R}^{\pm}_{\infty})^{\omega}$ *such that the restriction of* λ *to states, denoted* $\lambda|_Q$*, is a path in the model and the second component keeps track of the currently available resources; we use* $\lambda|_{\mathcal{R}}$ *to refer to the projection to the second component.*

3.2 Resource-Bounded Tree Logic

We present a logic based on $\mathsf{CTL}^{\star}$ which can be used to verify systems with limited resources. In the logic we replace the $\mathsf{CTL}^{\star}$ path quantifier E by $\langle\rho\rangle$ where ρ is a resource-quantity set. The intuitive reading of a formula $\langle\rho\rangle\gamma$ is that there is a(n) (infinite) ρ-feasible path λ on which γ holds. Note that E (there is a path in the system) can be defined as $\langle\rho^{\infty}\rangle$ where ρ^{∞} is the resource set assigning ∞ to each resource type. Formally, the language is defined as follows.

Definition 7 ($\mathcal{L}_{RTL^{\star}}$). *Let* $\mathcal{R}$ *be a set of resources and let* $\mathcal{P}rops$ *a set of propositions. The language* $\mathcal{L}_{RTL^{\star}}$ *is defined by the following grammar:*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\rho\rangle\gamma \text{ where } \gamma ::= \varphi \mid \neg\gamma \mid \gamma \wedge \gamma \mid \varphi\,\mathcal{U}\,\varphi \mid \bigcirc\varphi$$

and $\mathsf{p} \in \mathcal{P}rops$ *and* $\rho \in \mathcal{R}^{\pm}_{\infty}$*. Formulae* φ *(resp.* γ*) are called* state *(resp.* path*) formulae.*

Moreover, we define fragments of $\mathcal{L}_{RTL^\star}$ in which the domain of ρ is restricted. Let X be any set of multisets over $\mathcal{R}$. Then $\mathcal{L}_{RTL^\star_X}$ restricts $\mathcal{L}_{RTL^\star}$ in such a way that $\rho \in X$. Finally, we define $[\rho]$, the dual of $\langle\rho\rangle$, *as $\neg\langle\rho\rangle\neg$.*

Analogously to the language of **CTL** we define $\mathcal{L}_{RTL}$ as the fragment of $\mathcal{L}_{RTL^\star}$ in which each temporal operator is immediately preceded by a path quantifier.

Definition 8 ($\mathcal{L}_{RTL}$). *Let $\mathcal{R}$ be a set of resources and let $\mathcal{P}rops$ a set of propositions. The language $\mathcal{L}_{RTL}$ is defined by the following grammar:*

$$\varphi ::= \mathsf{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\rho\rangle\bigcirc\varphi \mid \langle\rho\rangle\Box\,\varphi \mid \langle\rho\rangle\varphi\,\mathcal{U}\,\varphi$$

where $\mathsf{p} \in \mathcal{P}rops$, $\rho \in \mathcal{R}^{\pm}_{\infty}$. Fragments $\mathbf{RTL}_X$ are defined in analogy to Def. 7.

As in the language of **CTL** we define $\Diamond\,\varphi$ as $\top\,\mathcal{U}\,\varphi$ and we use the following abbreviations for the universal quantifiers (they are not definable as duals in $\mathcal{L}_{RTL}$ as, for example, $\neg\langle\rho\rangle\neg\Box\,\varphi$ is not an admissible $\mathcal{L}_{RTL}$-formula):

$[\rho]\bigcirc\varphi \equiv \neg\langle\rho\rangle\bigcirc\neg\varphi$,
$[\rho]\Box\,\varphi \equiv \neg\langle\rho\rangle\Diamond\,\neg\varphi$,
$[\rho]\varphi\,\mathcal{U}\,\psi \equiv \neg\langle\rho\rangle((\neg\psi)\,\mathcal{U}\,(\neg\varphi \wedge \neg\psi)) \wedge \neg\langle\rho\rangle\Box\,\neg\psi$,

Next, we give the semantics for both languages.

Definition 9 (Semantics, RTL*). *Let $\mathfrak{M}$ be an* RBM, *let q be a state in $\mathfrak{M}$, and let $\lambda \in \Lambda_{\mathfrak{M}}$. The semantics of $\mathcal{L}_{RTL^\star}$-formulae is given by the satisfaction relation $\models$ which is defined as follows:*

$\mathfrak{M}, q \models \mathsf{p}$ *iff* $\lambda[0] \in \pi(\mathsf{p})$ *and* $\mathsf{p} \in \mathcal{P}rops$;
$\mathfrak{M}, q \models \varphi \wedge \psi$ *iff* $\mathfrak{M}, q \models \varphi$ *and* $\mathfrak{M}, q \models \psi$
$\mathfrak{M}, q \models \langle\rho\rangle\varphi$ *iff there is a ρ-feasible path* $\lambda \in \Lambda(q)$ *such that* $\mathfrak{M}, \lambda \models \varphi$
$\mathfrak{M}, \lambda \models \varphi$ *iff* $\mathfrak{M}, \lambda[0] \models \varphi$;

and for path formulae:

$\mathfrak{M}, \lambda \models \neg\gamma$ *iff not* $\mathfrak{M}, \lambda \models \gamma$
$\mathfrak{M}, \lambda \models \gamma \wedge \psi$ *iff* $\mathfrak{M}, \lambda \models \gamma$ *and* $\mathfrak{M}, \lambda \models \psi$
$\mathfrak{M}, \lambda \models \Box\,\varphi$ *iff for all* $i \in \mathbb{N}$ *we have that* $\lambda[i, \infty] \models \varphi$;
$\mathfrak{M}, \lambda \models \bigcirc\varphi$ *iff* $\lambda[1, \infty] \models \varphi$; *and*
$\mathfrak{M}, \lambda \models \varphi\,\mathcal{U}\,\psi$ *iff there is an* $i \geq 0$ *such that* $\mathfrak{M}, \lambda[i, \infty] \models \psi$ *and* $\mathfrak{M}, \lambda[j, \infty] \models \varphi$ *for all* $0 \leq j < i$;

We consider the logic **RTL*** *as the tuple $(\mathcal{L}_{RTL^\star}, \models)$ over all* RBMs *and analogously for all other fragments. These clauses are also used to define the semantics for $\mathcal{L}_{RTL}$ (therefore, we also stated the clause for $\Box\,\varphi$).*

Thus the meaning of $[\rho]\Box\,\mathsf{p}$ is that proposition p holds in every state on any ρ-feasible path.

We now discuss some interpretations of the formula $\langle\rho\rangle\gamma$ considering various resource-quantity sets. For $\rho \in \mathcal{R}^{\oplus}$ it is assumed that ρ consists of an initial (positive) amount of resources which can be used to achieve γ where the quantity of each resource is finite. $\rho \in \mathcal{R}^{\oplus}_{\infty}$ allows to *ignore* some resources (i.e., it is assumed that there is an infinite quantity of them). Initial debts of resources can be modelled by $\rho \in \mathcal{R}^{\pm}_{\infty}$.

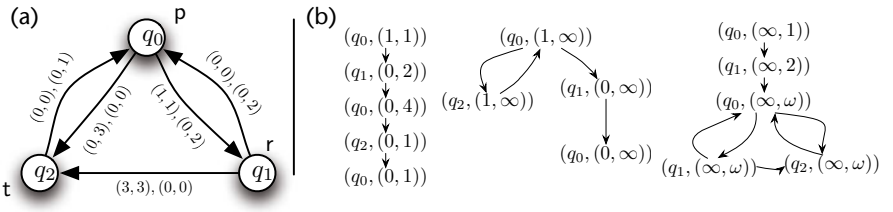

Fig. 1. In Figure (a) a simple RBM $\mathfrak{M}$ is shown and (b) presents some corresponding cover graphs

Example 1. Consider the RBM $\mathfrak{M}$ in Figure 1(a). Each transition is labeled by $(c_1, c_2), (p_1, p_2)$ with the interpretation: The transition consumes c_i and produces p_i quantities of resource r_i for $i = 1, 2$. We encode the resource-quantity set by (a_1, a_2) to express that there are a_i quantities of resoure r_i for $i = 1, 2$.

- If there are infinitely many resources available proposition t can become true infinitely often: $\mathfrak{M}, q_0 \models \langle(\infty,\infty)\rangle\Box \Diamond \mathsf{t}$
- We have $\mathfrak{M}, q_0 \not\models \langle(1,1)\rangle\Box \top$ as there is no $(1,1)$-feasible path. The formula $\langle(1,\infty)\rangle\Box (\mathsf{p} \vee \mathsf{t})$ holds in q_0.
- Is there a way that the system runs forever given specific resources? Yes, if we assume, for instance, that there are infinitely many resources of r_1 and at least one resource of r_2: $\mathfrak{M}, q_0 \models \langle(\infty,1)\rangle\top$

These simple examples show, that it is not always immediate whether a formula is satisfied, sometimes a rather tedious calculation might be required.

3.3 Cover Graphs and Cover Models

In this section we introduce a transformation of RBMs into Kripke models. This allows us, in general, to translate truth in **RTL** to truth in **CTL** as shown in Section 4.1.

We say that a resource-quantity set *covers* another, if it has at least as many resources of each type with at least one amount actually exceeding that of the other resource-quantity set. We are interested in cycles of transition systems that produce more resources than they consume, thereby giving rise to unbounded resources of some type(s). This is captured by a *cover graph* for RBMs, extending ideas from [11] and requiring an ordering on resource quantities.

Definition 10 (Resource ordering $<$). *Let ρ and ρ' be resource sets in $\mathcal{R}^{\pm}_{\infty}$. We say $\rho < \rho'$ iff $(\forall r \in \mathcal{R}\ (\rho(r) \leq \rho'(r))) \wedge (\exists r \in \mathcal{R}\ (\rho(r) < \rho'(r)))$. We say ρ has strictly less resources than ρ' or ρ'* covers *ρ.*

The ordering is extended to allow values of ω by defining for $x \in \mathbb{N}$ that $\infty + \omega = \infty$, $\infty - \omega = \infty$, $\omega - \infty = -\infty$, $\omega + x = \omega$, $\omega - x = \omega$, and $\omega < \infty$.

Definition 11 (ρ-feasible transition, $\xrightarrow{\rho}$). *We say that a transition $q \to q'$ is* ρ-feasible *and write $q \xrightarrow{\rho} q'$ if for all $r \in \mathcal{R}$ we have that $0 < {}^\bullet t_{q,q'}(r)$ implies ${}^\bullet t_{q,q'}(r) \leq \rho(r)$.*

So, given a specific amount of resources ρ a transition is said to be ρ-feasible if it can be traversed given ρ. A node of the cover graph consists of tuples $(q, (x_i)_{i=1,\ldots,|\mathcal{R}|})$ where q is a state of the RBM and $(x_i)_i$ is a vector representing the currently available resources. The variable x_i denotes that there are x_i units of resource r_i.

Definition 12 ((ρ, q)-cover graph of an RBM, path, $\lambda|_Q$). *Let $\mathfrak{M} = (Q, \to, \mathcal{P}rops, \pi, \mathcal{R}, t)$, let q be a state in Q, and let $\rho \in \mathcal{R}^{\pm}_{\infty}$. Without loss of generality, assume $\mathcal{R} = \{r_1, \ldots, r_n\}$ and consider $(x_i)_i$ as an abbreviation for the sequence $(x_i)_{i=1,\ldots,n}$. The* (ρ, q)-cover graph *$\mathcal{CG}(\mathfrak{M}, \rho, q)$ for $\mathfrak{M}$ with initial state $q \in Q$ and an initial resource-quantity set ρ is the graph (V, E) defined as the least fixed-point of the following specification:*

1. *$(q, (\rho(r_i))_i) \in V$ (the root vertex).*
2. *For $(q', (x_i)_i) \in V$ and $q'' \in Q$ with $q' \xrightarrow{(x_i)_i} q''$ then either:*
   (a) *if there is a vertex $(q'', (\hat{x}_i)_i)$ on the path from the root to $(q', (x_i)_i)$ in V, with $(\hat{x}_i)_i < (x_i - {}^\bullet t_{q',q''}(r_i) + t_{q',q''}{}^\bullet(r_i))_i$ then $(q'', (\tilde{x}_i)_i) \in V$ and $((q', (x_i)_i), (q'', (\tilde{x}_i)_i)) \in E$ where we define*

$$\tilde{x}_i := \begin{cases} \max\{\omega, x_i - {}^\bullet t_{q',q''}(r_i) + t_{q',q''}{}^\bullet(r_i)\} & \text{if } \hat{x}_i < x_i, \\ x_i - {}^\bullet t_{q',q''}(r_i) + t_{q',q''}{}^\bullet(r_i) & \text{otherwise;} \end{cases}$$

   (b) *or else $(q'', (x_i - {}^\bullet t_{q',q''}(r_i) + t_{q',q''}{}^\bullet(r_i))_i) \in V$ and $((q', (x_i)_i), (q'', (x_i - {}^\bullet t_{q',q''}(r_i) + t_{q',q''}{}^\bullet(r_i))_i)) \in E$.*

A path *in $\mathcal{CG}(\mathfrak{M}, \rho, q)$ is an infinite sequence of pairwise adjacent states. Given a path $\lambda = (q_1, (x_{1_i})_i)(q_2, (x_{2_i})_i) \ldots$ we use $\lambda|_Q$ to denote the path $q_1 q_2 \ldots$, i.e., the states of $\mathfrak{M}$ are extracted from the states in V.*

Cover graphs can be viewed as Kripke frames. It is obvious how they can be extended to models given an RBM.

Definition 13 ((ρ, q)-cover model of an RBM). *Let $G = (V, E)$ be the (ρ, q)-cover graph of an* RBM *$\mathfrak{M} = (Q, \to, \mathcal{P}rops, \pi, \mathcal{R}, t)$. The* (ρ, q)-cover model of $\mathfrak{M}$, *$\mathcal{CM}(\mathfrak{M}, \rho, q)$, is given by $(V, E, \mathcal{P}rops, \pi')$ with $\pi'((q, (x_i)_i)) := \pi(q)$ for all $(q, (x_i)_i) \in V$.*

Figure 2 shows the RBM $\mathfrak{M}$ in (a) and its cover model $\mathcal{CM}(\mathfrak{M}, 0, q_0)$ at the very top of (b). In the cover model, ω denotes the reachability of unbounded resources.

In Section 4.1 we analyse the relation between cover models and truth in **RTL**. Unfortunately, as illustrated in the next example, "simple" cover models in their current form are not yet suitable for that.

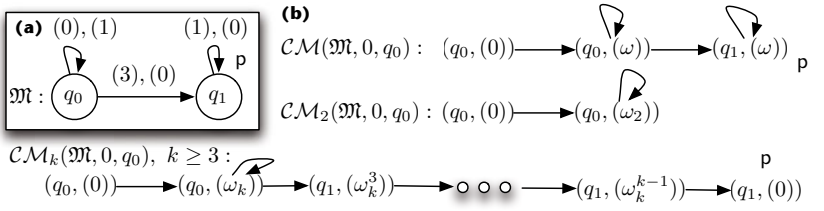

Fig. 2. An RBM $\mathfrak{M}$ (Fig. (a)), its cover model, 2-cover model, and κ-cover model (Fig. (b))

Example 2. Let λ be the path of $\mathcal{CM}(\mathfrak{M}, 0, q_0)$ from Figure 2(b) with $\lambda|_Q = q_0 q_0 (q_1)^\omega$. Obviously, this path is not 0-feasible in the model $\mathfrak{M}$ from Fig. 2(a). The problem is, that subsequent selections of the transition $q_0 \to q_0$ allows to generate any *finite* amount of resources, thus is covered by ω, but any finite amount is not enough for the subpath $(q_1)^\omega$. This implies, that we cannot directly use cover models as alternative models.

Note, however, that the following result is obvious by the definition of a cover model: Every ρ-feasible path in the model is also a path in the corresponding cover model. The other direction is the one that causes trouble.

Proposition 1. *If λ is a ρ-feasible q-path in $\mathfrak{M}$ then there is a (q, ρ)-path λ' in $\mathcal{CM}(\mathfrak{M}, \rho, q)$ such that $\lambda = \lambda'|_Q$.*

Proof. Let λ be a ρ-feasible q-path and η_i be the resources available at $\lambda[i]$ after $\lambda[0, i]$, for $i = 1, 2, \ldots$; particularly, we have that $\eta_0 = \rho$. By induction on the number of transitions we show that there is a (q, ρ)-path λ' in $\mathcal{CM}(\mathfrak{M}, \rho, q)$ where (V, E) denotes the underlying graph such that $\lambda = \lambda'|_Q$. By definition is $(\lambda[0], \rho(r_i)_i) \in V$. For every state q' with a ρ-feasible transition from $\lambda[0]$ to q' we have that $(q', \ldots) \in V$ and an edge $((\lambda[0], \rho), (q', \ldots)) \in E$ (according to the construction of the cover model). In particular, we have that $(\lambda[1], \zeta) \in V$ with $\zeta \geq_\omega \eta_1$.

Now suppose the claim is proven up to position k. Let $(\lambda[k], \zeta) \in V$ with $\zeta \geq_\omega \eta_k$ be the $k+1$st state on λ'. Following the same reasoning as above there is a transition $((\lambda[k], \zeta), (\lambda[k+1], \zeta')) \in E$ with $\zeta' \geq_\omega \eta_{k+1}$. □

In order to avoid the problem discussed in Example 2 we modify the cover graph construction as follows. The construction changes for those transitions that consume from the ω quantified resource type. Instead of using the rule "$\omega - k = \omega$", we (try to) expand the nodes for a fixed number of times ensuring that other loop's resource requirements can be met. But we abstain of introducing ω's as done in the cover graphs.

For the construction, we replace ω by κ new symbols ω_κ^l for $l = 0, \ldots \kappa - 1$ and $\kappa \in \mathbb{N}_0$. For $i \in \mathbb{N}_0$ we define: $\omega_\kappa^l - i = 0$ for $l + i \geq \kappa$, $\omega_\kappa^l - i = \omega_\kappa^{l+i}$ for $l + i < \kappa$, $\omega_\kappa^l + i = \omega_\kappa^{\min\{l-i,0\}}$, and we set $\omega_\kappa = \omega_\kappa^0$. The symbol ω_κ is used

to represent that at least κ units of some resource type are produced, and ω_κ^l indicates that there are $\kappa - l$ resources left.

Identifying the symbol ω_κ^l with the number $\kappa - l$ allows to extend the resource ordering from Definition 10 in a natural way; e.g. we have $i \leq \omega_\kappa^l$ iff $i \leq \kappa - l$. Moreover, this does also make it possible to lift the notation of ρ-feasible transition etc. to this extended case. Finally, we define a class of cover models.

Definition 14 ($\mathcal{CM}_\kappa(\mathfrak{M}, \rho, q)$)**.** *The construction of the* (ρ, q, κ)-cover graph *is defined as in Definition 12 but* ω *in 2. is replaced by* ω_κ*; that is,*

$$\tilde{x}_i := \begin{cases} \max\{\omega_\kappa, x_i - {}^\bullet t_{q',q''}(r_i) + t_{q',q''}{}^\bullet(r_i)\} & \text{if } \hat{x}_i < x_i, \\ x_i - {}^\bullet t_{q',q''}(r_i) + t_{q',q''}{}^\bullet(r_i) & \text{otherwise;} \end{cases}$$

The (ρ, q, κ)-cover model, $\mathcal{CM}_\kappa(\mathfrak{M}, \rho, q)$*, is defined analogously to Definition 13.*

In Figure 2(b) we have also drawn the 2- and κ-cover model of the model $\mathfrak{M}$. In the next example we show that this generalised cover models overcome the problem discussed in Example 2.

Example 3. The "bad" path λ of Example 2 is neither possible in $\mathcal{CM}_2(\mathfrak{M}, 0, q_0)$ nor in $\mathcal{CM}_\kappa(\mathfrak{M}, 0, q_0)$ for any $\kappa \geq 0$. This is, because for any fixed κ the path $(q_1)^\omega$ will eventually have consumed all resources from ω_κ.

However, another problem arises. If the κ is chosen too small then we might abort the construction too early. The cover model $\mathcal{CM}_2(\mathfrak{M}, 0, q_0)$ illustrates the problem: Principally, it is possible to reach state q_1 if the loop $q_0 \to q_0$ is traversed at least three times. However, as ω_2 does not allow to "remember" more than two units of resources state q_1 is never visited.

In order to avoid this problem we need to find an appropriate κ such that a theorem similar to Proposition 1 with respect to κ-cover models holds. Indeed, such a κ is constructible but it is very complex (cf. the proof of Theorem 3).

We end the section with two results.

Proposition 2. *Let* $\rho \in \mathcal{R}_\infty^\pm$*, let* $\mathfrak{M}$ *be an* RBM*, let* q *be a state in* $\mathfrak{M}$*, and let* G *denote the* (ρ, q)- *or* (ρ, q, κ) *cover graph of* $\mathfrak{M}$*. Then, for each node* $(q, (x_i)_i)$ *of* G *the property* $x_i \geq \min\{\rho(r_i), 0\}$ *holds.*

Proof. Suppose there is a node $(q, (x_i)_i)$ in the cover graph G and an index i such that $x_i < \min\{\rho(r_i), 0\}$. We first consider the case in which the minimum is equal to 0. Then, there must be a transition in G which causes a non-negative quantity of r_i to become negative. But such a transition is not feasible due to the construction of G! The case in which the minimum is equal to $\rho(r_i) < 0$ yields the same contradiction as a negative quantity of r_i reduces even further which is not allowed in the construction of G. □

The proposition states that non-positive resource quantities cannot decrease further. Theorem 1 states that cover models are finite; its proof is similar to the corresponding proof for Karp-Miller graphs [11].

Theorem 1 (Finiteness of the (κ-)cover graph). *Let $\rho \in \mathcal{R}^{\pm}_{\infty}$ and $\kappa \in \mathbb{N}$. The (ρ, q)- and (ρ, q, κ)-cover graphs of the* RBM $\mathfrak{M}$*, $q \in Q_{\mathfrak{M}}$, are finite.*

Proof. Let G denote the (ρ, q)-cover graph of $\mathfrak{M}$ and let Q be the set of states in $\mathfrak{M}$. Assume G is infinite (i.e., G has infinitely many nodes). Then, there is an infinite path $l = v_1 v_2 \dots$ in G that contains infinitely many different states. Since Q is finite there is some state, say $q' \in Q$, of $\mathfrak{M}$ and an infinite subsequence of distinct states $l' = v_{i_1} v_{i_2} \dots$ on l with $v_{i_j} = (q', (x^j_k)_k)$ and $i_j < i_{j+1}$ for all $j = 1, 2, \dots$. Due to the construction of the cover graph, it cannot be the case that $(x^j_k)_k \leq (x^{j'}_k)_k$ for any $1 \leq j < j'$; otherwise, an ω-node would have been introduced and the infinite sequence would have collapsed. So, there must be two distinct indices, o and p, with $1 \leq o, p \leq |\mathcal{R}|$ such that, without loss of generality, $x^j_o < x^{j'}_o$ and $x^j_p > x^{j'}_p$. But by Prop. 2 we know that each $x^j_k \geq \min\{\rho(r_k), 0\}$; hence, the previous property cannot hold for all indices o, p, j, j' but for the case in which $\rho(r) = -\infty$ for some resource r. However, this would also yield a contradiction as any non-negative resource quantity is bounded by 0. This proves that such an infinite path cannot exist and that the cover graph therefore has to be finite. □

3.4 Resource-Bounded Models

In Section 5 we show that the model-checking problem is decidable for **RTL**. Decidability of model checking for (full) **RTL*** over arbitrary RBMs is still open. However, we identify interesting subclasses in which the problem is decidable. Below we consider some restrictions which may be imposed on RBMs.

Definition 15 (Production free, zero (loop) free, k-bounded) *Let $\mathfrak{M} = (Q, \rightarrow, \mathcal{P}rops, \pi, \mathcal{R}, t)$ be an* RBM*.*

(a) We say that $\mathfrak{M}$ is production free *if for all $q, q' \in Q$ we have that $t_{q,q'} = (\mathbf{c}, \emptyset)$. That is, actions cannot produce resources they only consume them.*
(b) We say that $\mathfrak{M}$ is zero free *if there are no states $q, q' \in Q$ with $q \neq q'$ and $t_{q,q'} = (\emptyset, \mathbf{p})$. That is, there are no transitions between distinct states which do not consume any resources.*
(c) We say that $\mathfrak{M}$ is zero-loop free *if there are no states $q, q' \in Q$ with $t_{q,q'} = (\emptyset, \mathbf{p})$. That is, in addition to zero free models, loops without consumption of resources are also not allowed.*
*(d) We say that $\mathfrak{M}$ is (structurally) k-*bounded *for $\rho \in {}^k\mathcal{R}^{\pm}_{\infty}$ iff the available resources after any finite prefix of a ρ-feasible path are bounded by k, i.e., there is no reachable state in which the agent can have more than k resources of any resource type.*

In the following we summarise some results which are important for the model checking results presented in Section 5.

Proposition 3. *Let $\mathfrak{M}$ be an* RBM *and let $\rho \in \mathcal{R}^{\pm}_{\infty}$ be a resource-quantity set. Then, there is an* RBM *$\mathfrak{M}'$ and a $\rho' \in \mathcal{R}^{\pm}$, both effectively constructible from $\mathfrak{M}$ and ρ, such that the following holds: A path is ρ-feasible in $\mathfrak{M}$ if, and only if, it is ρ'-feasible in $\mathfrak{M}'$.*

Proof. Let ρ' be equal to ρ but the quantity of each resource r with $\rho(r) \in \{-\infty, \infty\}$ is 0 in ρ' and let $\mathfrak{M}'$ equal $\mathfrak{M}$ apart from the following exceptions. For each transition (q, q') with $t_{qq'} = (\mathbf{c}, \mathbf{p})$ in $\mathfrak{M}$ do the following: Set $\mathbf{c}(r) = 0$ in $\mathfrak{M}'$ for each r with $\rho(r) = \infty$; or remove the transition (q, q') completely in $\mathfrak{M}'$ if $\mathbf{c}(r) > 0$ (in $\mathfrak{M}$) and $\rho(r) = -\infty$ for some resource r. Obviously, $\rho \in \mathcal{R}^{\pm}$.

Now, the left-to-right direction of the result is straightforward as only transitions were omitted in $\mathfrak{M}'$ which can not occur on any ρ-feasible path in $\mathfrak{M}$. The right-to-left direction is also obvious as only resource quantities in $\mathfrak{M}'$ were set to 0 from which an infinite amount is available in ρ and only those transitions were removed which can never occur due to an infinite debt of resources. □

The next proposition presents some properties of special classes of RBMs introduced above. In general there may be infinitely many ρ-feasible paths. We study some restrictions of RBMs that reduce the number of paths:

Proposition 4. *Let* $\mathfrak{M} = (Q, \to, \mathcal{P}rops, \pi, \mathcal{R}, t)$ *be an* RBM.

(a) Let $\rho \in \mathcal{R}^{\pm}$ *and let* $\mathfrak{M}$ *be production and zero-loop free; then, there are no* ρ*-feasible paths.*
(b) Let $\rho \in \mathcal{R}^{\pm}$ *and let* $\mathfrak{M}$ *be production and zero free. Then, for each* ρ*-feasible path* λ *there is an (finite) initial segment* λ' *of* λ *and a state* $q \in Q$ *such that* $\lambda = \lambda' \circ qqq \ldots$.
(c) Let $\rho \in \mathcal{R}^{\pm}$ *and let* $\mathfrak{M}$ *be production free. Then, each* ρ*-feasible path* λ *has the form* $\lambda = \lambda_1 \circ \lambda_2$ *where* λ_1 *is a finite sequence of states and* λ_2 *is a path such that no transition in* λ_2 *consumes any resource.*
(d) Let $\rho \in \mathcal{R}^{\pm}$ *and let* $\mathfrak{M}$ *be* k*-bounded for* ρ*. Then there are only finitely many state/resource combinations (i.e., elements of* $Q \times \mathcal{R}^{\pm}$*) possible on any* ρ*-feasible path.*

Proof (Sketch).

(a) As there are no resources with an infinite amount and each transition is production free and consumes resources some required resources must be exhausted after finitely many steps.

(b) Apart from (a) loops may come for free and this is the only way how ρ-feasible paths can result.

(c) Assume the contrary. Then, in any infinite suffix of a path there is a resource-consuming transition that occurs infinitely often (as there are only finitely many transitions). But then, as the model is production free, the path cannot be ρ-feasible.

(d) We show that there cannot be infinitely many state/resource combinations reachable on any ρ-feasible path. Since the condition of ρ-feasibility requires the consumed resources to be present, there is no possibility of infinite decreasing sequences of resource-quantity sets.This gives a lower bound for the initially available resources ρ. The k-boundedness also gives an upper bound. □

We show that k-boundedness is decidable for RBMs.

Proposition 5 (Decidability of k-boundedness). *Given a model $\mathfrak{M}$ and an initial resource-quantity set ρ, the question whether $\mathfrak{M}$ is* k-bounded *for ρ is decidable.*

Proof. First, we check that $\rho \in {}^{k}\mathcal{R}_{\infty}^{\oplus}$. If this is not the case, then $\mathfrak{M}$ is not k-bounded for ρ. Then we construct the cover graph of $\mathfrak{M}$ and check whether there is a state $(q, (x_i)_i)$ in it so that $x_i > k$ for some i. If this is the case $\mathfrak{M}$ is not k-bounded; otherwise it is. □

We end this section with an easy result showing a sufficient condition for a model to be k-bounded.

Proposition 6. *Let $\rho \in \mathcal{R}^{\pm}$. Each production-free RBM is k-bounded for ρ where $k := \max\{i \mid \exists r \in \mathcal{R}\ (\rho(r) = i)\}$.*

4 Properties of Resource-Bounded Tree Logics

Before discussing specific properties of **RTL** and **RTL*** and showing the decidability of the model-checking problem for **RTL** and for special cases of **RTL*** and its models, we note that our logics conservatively extend **CTL*** and **CTL**. This is easily seen by defining the path quantifier E as $\langle \rho^\infty \rangle$ and by setting $t_{qq'} = (\emptyset, \emptyset)$ for all states q and q' where ρ^∞ denotes the resource set assigning ∞ to each resource type. Hence, every Kripke model has a canonical representation as an RBM. Moreover, given an RBM we can express the existence of a path (neglecting resources) by $\mathsf{E} := \langle \rho^\infty \rangle$. This allows to directly interpret **CTL** and **CTL*** formulae over RBMs.

Proposition 7 (Expressiveness). CTL* *(resp.* **CTL***) can be embedded in* **RTL*** *(resp.* **RTL***) over Kripke models and* RBMs*.*

Proof. Given a **CTL*** formula φ and a Kripke model $\mathfrak{M}$ we replace every existential path quantifier in φ by $\langle \rho^\infty \rangle$ and denote the result by φ'. Then, we extend $\mathfrak{M}$ to the canonical RBM $\mathfrak{M}'$ if it is not already an RBM and have that $\mathfrak{M}, q \models \varphi$ iff $\mathfrak{M}', q \models \varphi'$. □

4.1 RTL and Cover Models

We show that if there is a satisfying path in any κ-cover model; then, there also is a path in the corresponding RBM. Note however, this result does only hold for *positive* formulae of the kind $\langle \rho \rangle \gamma$.

Let λ be a finite sequence of states. Then, we recursively define λ^n for $n \in \mathbb{N}_0$ as follows: $\lambda^0 := \epsilon$ and $\lambda^i := \lambda^{i-1}\lambda$ for $i \geq 1$. That is, λ^n is the path which results from putting λ n-times in sequence.

The following lemma states that for flat $\mathcal{L}_{RTL}$-path formulae[1] it does not matter whether a cycle is traversed just once or many times. It can be proved by a simple induction on the path formula γ.

[1] A formula is said to be *flat* if it does not contain any path quantifier.

Lemma 1. *Let γ be an $\mathcal{L}_{RTL}$-path formula containing no more path quantifiers, let $\mathfrak{M}$ be an* RBM *and let λ be a path in $\mathfrak{M}$. Now, if $\tilde{\lambda} = q_1 \dots q_n$ is a finite subsequence of λ with $q_1 = q_n$ (note, that a single state is permitted as well), then, λ can be written as $\lambda_1 \tilde{\lambda} \lambda_2$ where λ_1, λ_2 are subsequences of λ and we have that : $\mathfrak{M}, \lambda \models \gamma$ if, and only if, $\mathfrak{M}, \lambda_1 \tilde{\lambda}^n \lambda_2 \models \gamma$ for all $n \in \{1, 2, \dots\}$.*

The second lemma states that one can always extend a path in the κ-cover model to a feasible path in the RBM by duplicating loops.

Lemma 2. *Let λ be a path in $\mathcal{CM}_\kappa(\mathfrak{M}, \rho, q), (q, \rho)$ and $\lambda' = \lambda|_Q$; then, there are tuples $(a_i, b_i, c_i) \in \mathbb{N}_0^2 \times \mathbb{N}$ for $i = 1, 2, \dots$ such that for all $j = 1, 2, \dots$ we have that $a_j \leq b_j < a_{j+1}$ and $\lambda'[a_j] = \lambda'[b_j]$ and the path*

$$(\lambda'[a_i, b_i]^{c_i})_{i=1,2,\dots} \text{ is } \rho\text{-feasible in } \mathfrak{M}.$$

Proof. Let a $(q, (\rho(r_i))_i)$-path $\lambda = l_1 l_2 \dots$ in $G := \mathcal{CM}_\kappa(\mathfrak{M}, \rho, q) = (V, E)$ be given. We extend λ to a path λ' (having the structure as stated in the lemma) such that $\lambda'|_Q$ is ρ-feasible in $\mathfrak{M}$.

If $\lambda|_Q$ is ρ-feasible we just take λ' as λ. So, suppose $\lambda|_Q = q_{i_1} q_{i_2} \dots$ is not ρ-feasible. Then, there is a transition in λ that is not feasible in $\mathfrak{M}$. Let $l_1 \dots l_{k+1}$ be the *minimal* length initial subpath of λ such that $(l_1 \dots l_{k+1})|_Q$ is not feasible in $\mathfrak{M}$ and let $l_k = (q, (x_i)_i)$. According to the construction of cover graphs this can only be caused by a resource r_l such that $x_l = \omega_\kappa^t$ for $0 \leq t \leq \kappa$. Let $l_o = (q', (x'_i)_i)$ with $1 \leq o \leq k$ and o maximal be the state on λ at which x'_l was set to ω_κ most recently. Then, there must be another state $l_p = (q', (x''_i)_i)$, $1 \leq p < o$ and p maximal, with $(x''_i)_i < (x'_i)_i$ and $x''_l < x'_l$. The setting is depicted in Figure 3.

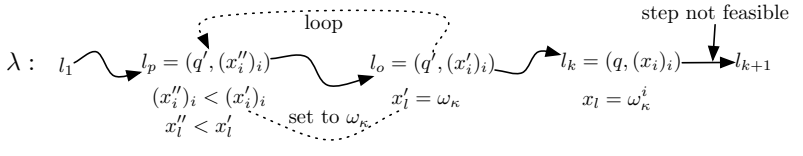

Fig. 3. Proof of Lemma 2

So, we extend λ to λ' by duplicating the subsequence $l_p l_{p+1} \dots l_o$ in l and adjusting the resources of the states preceding l_p accordingly. Thus, we have that $\lambda'|_Q = q_{i_1} \dots q_{i_p} q_{i_{p+1}} \dots q_{i_o} q_{i_p} \dots q_{i_o} q_{i_{o+1}} \dots$. We subsequently continue this procedure (now applied to λ') and do only duplicate transitions that are also present in λ (i.e. not the new ones). It remains to show that this procedure does not force some c_i to become infinite.

Suppose that there is some c_i that becomes infinite following this construction. Then, there is a set of resources that requires the resources produced by $\lambda[a_i, b_i]$; and there is no other loop (or set of loops) that starts after $\lambda[b_i]$ that would also provide the needed resources (otherwise these loops would be duplicated as the

construction looks for the latest possibility). In a κ-cover model, however, one can only "remember" κ units of a resource; hence, one can have at most κ transitions consuming of a specific resource until some other transition has to produce this very resource again. Thus, in order to ensure that λ is a path in G there must be a producing transition after $\lambda[b_i]$, in particular, a cycle introducing another ω_κ-node following the same line of argumentation as above, which contradicts our supposition. Hence, we will actually obtain a path λ' such that $\lambda'|_Q$ is ρ-feasible and has the structure $(\lambda|_Q[a_i, b_i]^{c_i})_{i=1,2,\ldots}$. □

Theorem 2. *Let $\rho \in \mathcal{R}^{\pm}_{\infty}$, let $\mathfrak{M}$ be an* RBM*, let q be a state in $\mathfrak{M}$. Then, for any κ and any flat $\mathcal{L}_{RTL}$-formula $\langle\rho\rangle\gamma$ we have that:*

$$\textit{If } \mathcal{CM}_\kappa(\mathfrak{M}, \rho, q), (q, \rho) \models \mathsf{E}\gamma \textit{ then } \mathfrak{M}, q \models \langle\rho\rangle\gamma.$$

Proof. The result follows from Lemma 1 and 2. Firstly, the path λ is extended to a path λ' such that $\lambda'|_Q$ is ρ-feasible according to Lemma 2; then, Lemma 1 shows that the truth of the flat path formula according to λ' does not change. □

Remark 1. Note, that the proof of Theorem 3 gives an algorithm that particularly allows to construct a fixed index κ from an RBM and $\langle\rho\rangle\gamma$ such that the "reverse" of Theorem 2 holds: If $\mathcal{CM}_\kappa(\mathfrak{M}, \rho, q), (q, \rho) \models \mathsf{E}\gamma$ then $\mathfrak{M}, q \models \langle\rho\rangle\gamma$. This construction of κ however does already "solve" the model checking problem and is computationally very expensive.

4.2 RTL* and Bounded Models

The case for **RTL*** is more sophisticated as the language is able to characterise more complex temporal patterns. It is still open whether the general case is decidable. In the following, we discuss the effects of various properties of RBMs with respect to **RTL***. For a given resource quantity it is possible to transform a structurally k-bounded RBM into a production-free RBM such that satisfaction of specific path formulae is preserved.

Proposition 8. *Let $\rho \in \mathcal{R}^{\pm}$, let $\mathfrak{M}$ be a structurally k-bounded* RBM *for ρ, and let q be a state in $\mathfrak{M}$. Then, we can construct a finite, production-free* RBM *$\mathfrak{M}'$ such that for every $\mathcal{L}_{RTL^\star}$-path formula γ containing no more path quantifiers the following holds:*

$$\mathfrak{M}, q \models \langle\rho\rangle\gamma \quad \textit{if, and only if,} \quad \mathfrak{M}', q' \models \langle\emptyset\rangle\gamma.$$

Proof (Sketch). We essentially take $\mathfrak{M}'$ as the reachability graph of $\mathfrak{M}$. This graph is build similar to the cover graph but no ω-nodes are introduced. Because there are only finitely many distinct state/resource combinations in $\mathfrak{M}$ (Prop. 4) the model is finite and obviously also production free.

Let $\mathfrak{M}, q \models \langle\rho\rangle\gamma$ and let λ be a ρ-feasible path satisfying γ. Then, the path obtained from λ by coupling each state with its available resources is a path in $\mathfrak{M}'$ satisfying γ. Conversely, let λ be a path in $\mathfrak{M}'$ satisfying γ. Then, $\lambda|_Q$ is a γ satisfying ρ-feasible path in $\mathfrak{M}$ due to the construction of $\mathfrak{M}'$. □

The following corollary is needed for the model-checking results in Section 5.

Corollary 1. *Let $\rho \in \mathcal{R}^{\pm}$, let $\mathfrak{M}$ be a structurally k-bounded* RBM *for ρ, and let q be a state in $\mathfrak{M}$. Then, we can construct a finite Kripke model such that for every $\mathcal{L}_{RTL^\star}$-path formula γ containing no more path quantifiers the following holds:*

$$\mathfrak{M}, q \models \langle\rho\rangle\gamma \quad \textit{if, and only if,} \quad \mathfrak{M}', q' \models \mathsf{E}\gamma.$$

Lemma 3 states that loops that do not consume resources can be reduced to a fixed number of recurrences. For a path λ, we use $\lambda^{[n]}$ to denote the path which is equal to λ but each subsequence of states $q_1 q_2 \dots q_k q$ occurring in λ with $q' := q_1 = q_2 = \dots = q_k \neq q$ and $k > n$ where the transition $q' \to q'$ does not consume any resource (i.e. the first k states represent a consumption-free loop that is traversed k times) is replaced by $q_1 q_2 \dots q_n q$. That is, states $q_{n+1} q_{n+2} \dots q_k$ are omitted. Note, that $\lambda^{[n]}$ is also well-defined for pure Kripke models.

Lemma 3. *(a) Let $\mathfrak{M}$ be a Kripke model and γ be a path formula of* **CTL*** *containing no path quantifiers and length $|\gamma| = n$. For every path λ in $\Lambda_{\mathfrak{M}}$ we have that $\mathfrak{M}, \lambda \models \gamma$ if, and only if, $\mathfrak{M}, \lambda^{[n]} \models \gamma$.*

(b) Let $\mathfrak{M}$ be a production- and zero-free RBM *and γ be an $\mathcal{L}_{RTL^\star}$-path formula containing no path quantifiers and length $|\gamma| = n$. Then, for each path λ in $\Lambda_{\mathfrak{M}}$ the following holds true: $\mathfrak{M}, \lambda \models \gamma$ if, and only if, $\mathfrak{M}, \lambda^{[n]} \models \gamma$.*

Note that we might want to allow to re-enter loops n-times for cases in which the formula has the form $\bigcirc\bigcirc \dots \bigcirc\Diamond\, \varphi$.

5 Model Checking Resource-Bounded Tree Logic

We are mainly interested in the verification of systems. *Model checking* refers to the problem whether a formula φ is true in an RBM $\mathfrak{M}$ and a state q in $\mathfrak{M}$. For **CTL*** this problem is **PSPACE**-complete and for **CTL**, the fragment of **CTL*** in which every temporal operator is directly preceded by a path quantifier, it is **P**-complete [7]. So, we cannot hope for our problem to be computationally any better than **PSPACE** in the general setting; actually, it is still open whether it is decidable at all.

The following result shows that model checking **RTL** is decidable.

Theorem 3 (Model Checking RTL: Decidability). *The model-checking problem for* **RTL** *over* RBMs *is decidable.*

Proof (Idea). A more elaborated proof sketch can be found in Appendix A. The main idea is to encode an RBM as a Petri net and then use decision procedures for Petri nets, more precisely a variant of the reachability problem. □

In the following, we consider the decidability of fragments of the full logic over special classes of RBMs (which of course, implies decidability of the restricted version over the same class of models).

Proposition 9 (Decidability: Production -, zero free). *The model-checking problem for* $\mathsf{RTL}^{\star}{}_{\mathcal{R}^{\pm}}$ *over production- and zero-free* RBMs *is decidable.*

Proof (Sketch). According to Prop. 4 and Lemma 3 there are only finitely many ρ-feasible paths of interest for $\rho \in \mathcal{R}^{\pm}$. This set can be computed step by step. Then, for $\mathfrak{M}, q \models \langle\rho\rangle\gamma$ where γ is a path formula one has to check whether γ holds on one of these finitely many ρ-feasible paths starting in q. The model checking algorithm proceeds bottom-up. □

From Corollary 1 we know that we can use a $\mathsf{CTL}^{\star}$ model checker over k-bounded models.

Proposition 10 (Decidability: k-bounded). *The model-checking problem for* $\mathsf{RTL}^{\star}{}_{\mathcal{R}^{\pm}}$ *over k-bounded* RBMs *is decidable and* **PSPACE***-hard.*

By Prop. 6 and the observation that resources with an infinite quantity can be neglected in a production-free RBM we can show the following theorem.

Theorem 4 (Decidability: production free). *The model-checking problem for* $\mathsf{RTL}^{\star}$ *over production-free* RBMs *is decidable and* **PSPACE***-hard.*

6 Conclusions, Related and Future Work

In this paper we have introduced resources into $\mathsf{CTL}^{\star}$ [6], which is arguably among the most important logics for computer science. The paper showed decidability results in the presence of some limiting constraints on the resource allocation for transitions in Kripke models.

While most agent models do not come with an explicit notion of resources, there is some recent work that take resources into account. [12] considers resources in conjunction with reasoning about an agent's goal-plan tree. Time, memory, and communication bounds are studied as resources in [2]. In [1] the abilities of agents under bounded memory are considered. Instead of asking for an arbitrary winning strategy a winning strategy in their setting has to obey given memory limitations.

A detailed analysis of the model checking complexity and the decidability question for the general case is left for future research. We are particularly interested in finding constraints that would make the extended logic's model-checking problem *efficiently* decidable for a relevant class of MAS.

Moreover, we are interested in the reasoning about and modelling of abilities of *multiple* agents having limited resources at their disposal. In [5] we consider an extension of the resource-bounded setting introduced here in the context of multi-agent systems (influenced by **ATL** [4] a logic for reasoning about strategic abilities of agents). In that paper we show that the problem is undecidable in general. On the other hand, if productions of resources are not allowed (as in [2]) it was recently shown that the model checking problem is decidable [3]. The authors of [3] do also propose a sound and complete axiomatisation of their

resource-based extension of **ATL** (the logic is called *resource-bounded alternating-time temporal logic*).

Another direction is offered by Linear Logic. Although Girard's linear logic [9] is not directly suitable for model checking, we will be looking into possible combinations of linear logic fragments with our approach. One idea is to formalise resources and their production/consumption by means of linear logic formulae and hope to come up with an axiomatisation for our logic.

References

1. Ågotnes, T., Walther, D.: A logic of strategic ability under bounded memory. J. of Logic, Lang. and Inf. 18(1), 55–77 (2009)
2. Alechina, N., Logan, B., Nga, N.H., Rakib, A.: Verifying time, memory and communication bounds in systems of reasoning agents. In: AAMAS 2008: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 736–743 (2008)
3. Alechina, N., Logan, B., Nga, N.H., Rakib, A.: Resource-bounded alternating-time temporal logic. In: van der Hoek, W., Kaminka, G., Lespérance, Y., Luck, M., Sen, S. (eds.) Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, IFAAMAS (to appear, May 2010)
4. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. Journal of the ACM 49, 672–713 (2002)
5. Bulling, N., Farwer, B.: On the (Un-)Decidability of Model-Checking Resource-Bounded Agents. In: Coelho, H., Wooldridge, M. (eds.) Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010), Porto, Portugal, August 16-20 (to appear, 2010)
6. Clarke, E.M., Emerson, E.A.: Design and synthesis of synchronization skeletons using branching time temporal logic. In: Kozen, D. (ed.) Logic of Programs 1981. LNCS, vol. 131, pp. 52–71. Springer, Heidelberg (1982)
7. Clarke, E.M., Emerson, E.A., Sistla, A.P.: Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Transactions on Programming Languages and Systems 8(2), 244–263 (1986)
8. Emerson, E.A., Halpern, J.Y.: Sometimes and not never revisited: On branching versus linear time temporal logic. In: Proceedings of the Annual ACM Symposium on Principles of Programming Languages, pp. 151–178 (1982)
9. Girard, J.-Y.: Linear logic. Theoretical Computer Science 50, 1–102 (1987)
10. Jančar, P.: Decidability of a temporal logic problem for petri nets. Theor. Comput. Sci. 74(1), 71–93 (1990)
11. Karp, R.M., Miller, R.E.: Parallel program schemata. Journal of Computer and System Sciences 3(2), 147–195 (1969)
12. Shaw, P., Farwer, B., Bordini, R.: Theoretical and experimental results on the goal-plan tree problem (short paper). In: Proceedings of AAMAS 2008, pp. 1379–1382 (2008)

A Proof of the RTL Model Checking Result

Theorem 5. *The model-checking problem for* $\mathsf{RTL}_{\mathcal{R}^{\oplus}_{\infty}}$ *over* RBMs *is decidable.*

Proof (Sketch). Firstly, we present the proof for feasible resource sets only. Proposition 3 allows to focus on resource-quantity sets from $\mathcal{R}^{\oplus}$. The main idea is to encode an RBM as a Petri net and then use decision procedures for Petri nets to solve the model checking problem. A Petri net is a tuple $N = (S, T, W, m^I)$ where S and T are non-empty and disjoint sets of places and transitions, $W : (S \times T) \cup (T \times S) \to \mathbb{N}_0$ represents arc weights that determine how many tokens are needed by and how many tokens are produced by each transition. Finally, $m^I : P \to \mathbb{N}_0$ is the initial marking, i.e., a distribution of tokens on the places of the net. A transition t is said to be enabled in a marking $m : P \to \mathbb{N}_0$) if $m(s) \geq W(s,t)$ for all $s \in S$. In this case, we also say, that t is m-enabled. Now, an m-enabled marking t may *fire* resulting in a new marking $m' := m - W(\cdot, t) + W(t, \cdot)$. Recursively, one defines the change that occurs given a sequence σ of subsequently fired transitions; thus, a *run* is an infinite sequence of subsequently enabled and firing transitions.

Now, we can encode an RBM $\mathfrak{M}$ with respect to a given set $Q' \subseteq Q_{\mathfrak{M}}$, and a feasible resource set ρ as a Petri net $N_{Q',\rho}(\mathfrak{M}) = (S, T, W, m^I)$. The main idea of encoding transitions is sketched in Figure 4. States q are encoded as places p_q and transitions between states as transitions between places. For each resource type a new place is created. For the initial marking function m^I we have that $m^I(p_q) = 1$ for all $q \in Q'$, $m^I(r) = \rho(r)$ for $r \in \mathcal{R}$, and 0 otherwise. A complete encoding of an RBM is shown in Figure 5. We denote (the unique) transition between place p_{q_i} and p_{q_j} by $t_{q_i q_j}$. (Note, that we are economical with our notation and reuse t already known from RBMs.)

Lemma 4. *Let* ρ *be a feasible resource set,* $\mathfrak{M}$ *an* RBM*, and* $q \in Q_{\mathfrak{M}}$*. Then, the following holds:*

$q_0 q_1 \ldots$ *is a* ρ*-feasible path in* $(\mathfrak{M}, q)$ *iff* $\sigma = t_{q_0 q_1} t_{q_1 q_2} \ldots$ *a run in* $N_{\{q_0\},\rho}(\mathfrak{M})$.

In order to model check specific formulae, we need to extend our encoding. For example, consider the formula $\langle\rho\rangle\Diamond\ \varphi$ where φ is a propositional formula and ρ

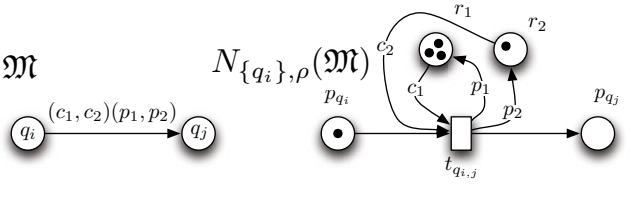

Fig. 4. Petri-net encoding $N_{\{q_i\},\rho}(\mathfrak{M})$ of an RBM $\mathfrak{M}$. Tokens inside the places r_k represent the amount of that resource (i.e., $\rho(r_1) = 3$ and $\rho(r_2) = 1$). Outgoing paths consume tokens and incoming paths produce tokens, labeled edges produce/consume the amount the edge is annotated with. E.g., if there is a token in place p_{q_i} and c_k tokens in place r_k then the token can be moved to p_{q_j} and p_k tokens can be moved to r_k for $k = 1, 2$.

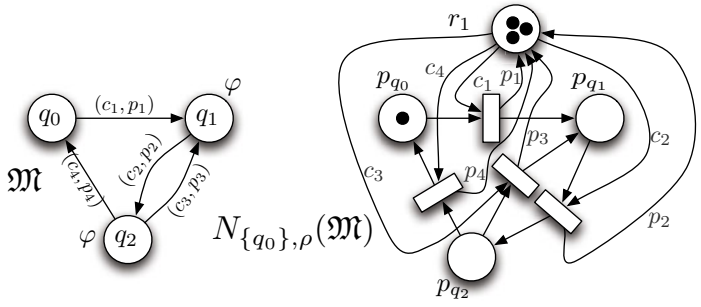

Fig. 5. Example of a complete encoding of an RBM $\mathfrak{M}$ where $\rho(r_1) = 3$

a feasible resource set. We can decompose the model checking problem into two parts:

1. Find a (finite) sequence of states feasible given ρ to a state in which φ holds; and
2. then arbitrarily extend this (finite) sequence to an infinite ρ-feasible path.

To achieve this, we introduce a new place that tells us (by marking it with a token) that φ has been made true. This place remains marked throughout the subsequent executions of the net and hence serves as an indicator of item 1 having been satisfied. To achieve this, given a propositional formula φ we extend the encoding $N_{\{q_0\},\rho}(\mathfrak{M})$ of $\mathfrak{M}$ to an encoding $N_{\{q_0\},\rho}(\mathfrak{M}, Q', \varphi)$ where $Q' \subseteq Q$ as explained in the following. The new Petri net is equal to $N_{\{q_0\},\rho}(\mathfrak{M})$ apart from the following modifications (Figure 6 illustrates the construction):

1. N' has two new places p_S and p_φ.
2. For each transition t in $N(\mathfrak{M})$ that corresponds to a transition $q \to q'$ in $\mathfrak{M}$ such that $q \in Q'$ and $q' \models^{\text{prop}} \varphi$ we construct a duplicate with the fresh name $\hat{t}$ and include the following arcs: p_S is connected to t; t and $\hat{t}$ are connected to p_φ; and p_φ is also connected to $\hat{t}$; i.e. $W(p_S, t) = W(t, p_\varphi) = W(p_\varphi, \hat{t}) = W(\hat{t}, p_\varphi) = 1$.
3. p_S is initially marked.

The constructed Petri net $N_{\{q_0\},\rho}(\mathfrak{M}, \{q_0\}, \varphi)$ has the following properties.

Proposition 11

1. *A transition t can only be enabled if there is a token in p_S.*
2. *Once such a transition t has fired it can never be enabled again and there is a token in p_φ*
3. *A transitions $\hat{t}$ can only be enabled if there is a token in p_φ.*
4. *Once there is a token in p_φ it remains there forever.*
5. *p_S and p_φ contain at most one token and there is a token in p_S iff there is no token in p_φ.*

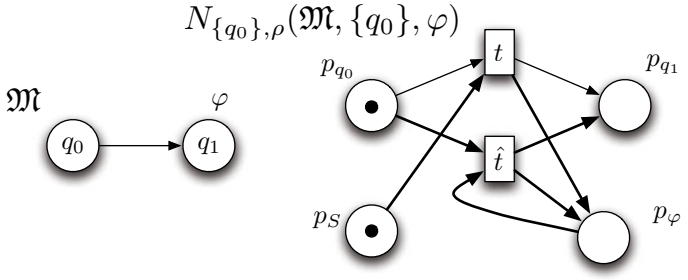

Fig. 6. The encoding $N_{\{q_0\},\rho}(\mathfrak{M}, \{q_0\}, \varphi)$ of an RBM $\mathfrak{M}$. The resource requirements are left out here.

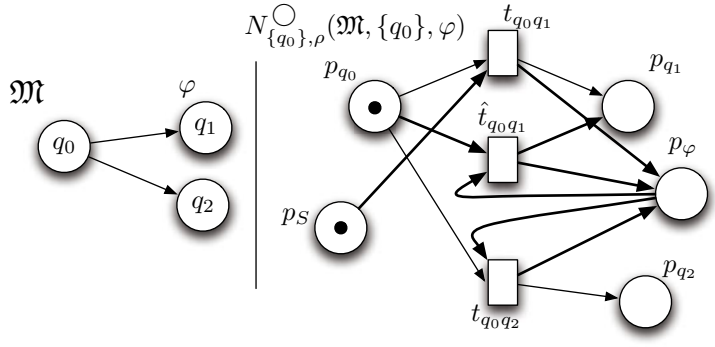

Fig. 7. The encoding $N^{\bigcirc}_{\{q_0\},\rho}(\mathfrak{M}, \{q_0\}, \varphi)$ of an RBM $\mathfrak{M}$. The resource requirements are left out here.

Additionally, for the next-operator we extend the construction and disable, in the first step, transition that do not result in a state satisfying φ. These transition are only enabled if there is a token in p_φ. The net is shown in Figure 7.

The next lemma provides the essential step to use decision procedures for Petri nets in order to solve the model checking problem.

Lemma 5

(a) $\mathfrak{M}, q_0 \models \langle\rho\rangle\Diamond\ \varphi$ *iff there is a run in* $N^{\Diamond}$ *on which there is a token in* p_φ *at some moment where* $N^{\Diamond}$ *is the Petri net that equals* $N_{\{q_0\},\rho}(\mathfrak{M}, Q_{\mathfrak{M}}, \varphi)$ *with the exception that the initial token in* p_S *is in* p_φ *instead iff* $q_0 \models^{prop} \varphi$.

(b) $\mathfrak{M}, q_0 \models \langle\rho\rangle\bigcirc\varphi$ *iff there is a run in* $N^{\bigcirc}_{\{q_0\},\rho}(\mathfrak{M}, \{q_0\}, \varphi)$ *on which there is a token in* p_φ *at some moment.*

(c) $\mathfrak{M}, q_0 \models \langle\rho\rangle\Box\ \varphi$ *iff there is a run in* $N^{\Box}$ *on which there never is a token in* $p_{\neg\varphi}$ *where* $N^{\Box}$ *is the Petri net that equals* $N_{\{q_0\},\rho}(\mathfrak{M}, Q_{\mathfrak{M}}, \neg\varphi)$ *with the exception that the initial token in* p_S *is in* $p_{\neg\varphi}$ *instead iff* $q_0 \not\models^{prop} \varphi$.

It remains to link the "until" case to Petri nets. For this, we consider the problem whether $\mathfrak{M}, q_0 \models \langle\rho\rangle\varphi\,\mathcal{U}\,\psi$. Let $\mathfrak{M}^{\varphi}$ be the restriction of $\mathfrak{M}$ to states in which φ holds. Now, $\mathfrak{M}^{\mathcal{U}}$ is the model that glues together $\mathfrak{M}^{\varphi}$ with $\mathfrak{M}$ as follows: Every state q in $\mathfrak{M}^{\varphi}$ is connected to a state $q' \in \mathfrak{M}$ if $q \to_{\mathfrak{M}} q'$ and q' satisfies ψ. The construction is illustrated in Figure 8.

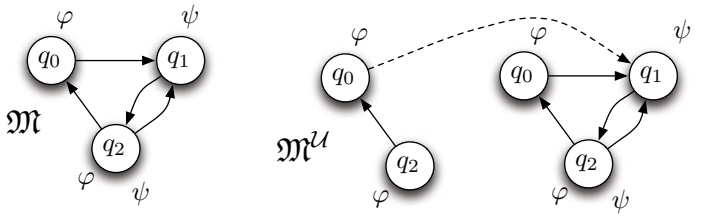

Fig. 8. Extending the RBM $\mathfrak{M}$ to $\mathfrak{M}^{\mathcal{U}}$ for $\varphi\,\mathcal{U}\,\psi$

Lemma 6. *Suppose $q_0 \models^{prop} \varphi$ (the other cases are trivially decidable). $\mathfrak{M}, q_0 \models \langle\rho\rangle\varphi\,\mathcal{U}\,\psi$ iff there is a run in $N^{\mathcal{U}}$ on which there is a token in p_ψ at some moment where $N^{\mathcal{U}}$ is the Petri net that equals $N_{\{q_0\},\rho}(\mathfrak{M}^{\mathcal{U}}, Q_{\mathfrak{M}^{\mathcal{U}}}, \psi)$ with the exception that the initial token in p_S is in p_ψ instead iff $q_0 \models^{prop} \psi$.*

Proof (of Lemma). The construction ensures that only states satisfying φ are visited until a state ψ is visited. The rest follows from Lemma 5(a). □

Finally, we show that the Petri net part in the previous two lemmata can be decided. Let a Petri net N and a pair (A, f) such that $A \subseteq S$ and $f : A \to \mathbb{N}_0$ be given. In [10] the following problem, here denoted by *ExtReach*, was shown to be decidable:

> Is there a run $\sigma = t_1 t_2 \ldots$ where t_1 is enabled by the initial marking $m_0 = m^I$ and firing t_1 leads to the successor marking m_1 and m_j enables t_{j+1} whose fireing leads to m_{j+1} for all $j > 1$, such that there are infinitely many indices i such that the marking m_i that occurs after t_i restricted to the states in A equals f (i.e., $m_i|_A = f$ for infinitely many i)?

We have the following reductions.

Lemma 7. *Assume the same notation as in Lemma 5 and 6.*

(a) There is a run in $N^{\bigcirc}$ on which there is a token in p_φ at some moment iff $(N^{\bigcirc}, (\{p_\varphi\}, f_1))$ is in ExtReach where f_1 is the constant function 1.
(b) There is a run in $N^{\Diamond}$ on which there is a token in p_φ at some moment iff $(N^{\Diamond}, (\{p_\varphi\}, f_1))$ is in ExtReach where f_1 is the constant function 1.
(c) There is a run in $N^{\Box}$ on which there never is a token in $p_{\neg\varphi}$ iff $(N^{\Box}, (\{p_{\neg\varphi}\}, f_0))$ is in ExtReach where f_0 is the constant function 0.
(d) There is a run in $N^{\mathcal{U}}$ on which there is a token in p_ψ at some moment iff $(N^{\mathcal{U}}, (\{p_\psi\}, f_1))$ is in ExtReach where f_1 is the constant function 1.

Proof (of Lemma). (a) The following follows from Proposition 11. There is a run on which there is a token in p_φ at some moment iff there is a run on which there is a token in p_φ infinitely often iff there is a run on which there is exactly one token in p_φ infinitely often iff $(N^{\bigcirc}, (\{p_\varphi\}, f_1))$ is in *ExtReach*.

(b-d) These cases are handled analogously. □

The *ExtReach* problem is solved by applying the reachability problem for Petri nets. If a marking is reachable an appropriate sequence of transitions is constructed. This sequence can also be used to construct κ: One simply takes the maximum of all markings of all resource types along this sequence. If the state is not reachable, κ is chosen arbitrarily. □

Finally, we also include non-feasible resource sets and get the main result.

Theorem 6 (Model CheckingRTL: Decidability). *The model-checking problem for* **RTL** *over* RBMs *is decidable.*

Proof (Sketch). We extend the previous construction to be able to deal with non-feasible resource sets.

For non-feasible initial resource sets, we can still have a feasible path, in case no resources with negative amount are ever required in the run (note that such resources can still be produced!).

We encode a non-feasible resource set by splitting each resource place r of the Petri net into a place for a positive number of resources, r, and a place for a negative number of resources, r^-.

Further, we need to ensure in our net, that whenever resources are produced a positive number of tokens is placed on the positive resource place (only if no tokens are present in the negative resource place) or a number of tokens is removed from the negative resource place. Combinations are possible, if the number of resources produced is larger than the negative number of resources currently available. In the latter case all resources are removed from the negative resource place and the remaining difference is placed into the positive place. Therefore, we introduce a special resource control state, r^{ctrl}, that "deactivates" the new part of the construction once a non-negative amount of resources is available.

In the following we will describe the construction in detail. Consider the transition of an RBM at the left-hand side of Figure 9. For simplicity, we only consider a single resource-type r. The transition consumes zero units of r and produces u units (note, that if the transition does also consume of this resource type we take the standard construction from Theorem 5). Suppose, we would like to model check a formula $\langle\rho\rangle\gamma$ with $\rho(r) = -d$, that is, there is an initial debt of d units of resource r. Firstly, we add a transition $t_{q_i q_j}$ from p_{q_i} to q_{q_j} which is only enabled if there are d units in the resource control state r^{ctrl} and a token on p_{q_i}. We add u transitions $t^1, \ldots, t^u$; $u-1$ places $p^1, \ldots, p^{u-1}$; and $u-1$ intermediate transitions $t^{p^1}, \ldots, t^{p^{u-1}}$. Their connections are shown in the right-hand part of Figure 9. Each transition t^i can only be enabled if there is a debt of resources (i.e. tokens in r^-). Such a transition takes one token from r^-

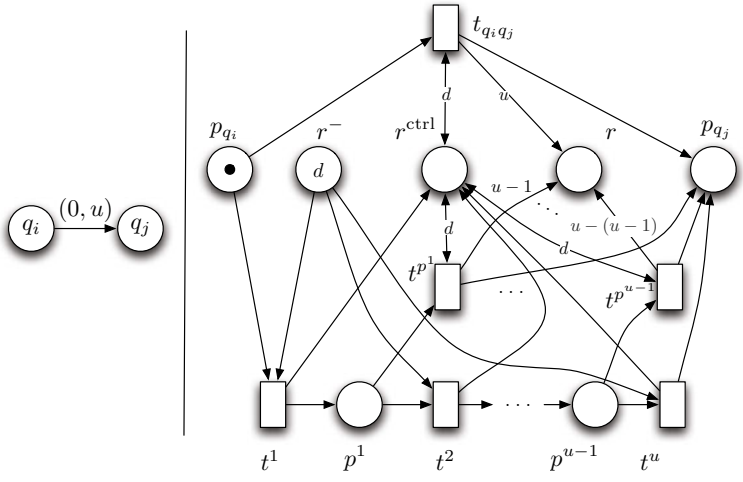

Fig. 9. Example of a PN construction for non-feasible resource sets: The left-hand RBM with a single resource r with $\rho(r) = -d$ is converted to the right-hand PN

and moves it to the control state r^{ctrl}. Once, there are d tokens in the control state the transitions t^{p^i} can be enabled (while t^i can no longer be enabled) and the remaining produced resources are added to the resource place r. The net has the following properties.

Proposition 12

1. *There are x tokens in r^- iff there are $d-x$ token in r^{ctrl} for $x \in \{0, \ldots, d\}$. (That is, r^- and r^{ctrl} are complementary places.)*
2. *Transitions $t_{q_iq_j}$ and $t^{p^1}, \ldots, t^{p^{u-1}}$ can only fire if there are d tokens in r^{ctrl}.*
3. *The number of tokens in r^{ctrl} is bounded by d and it is monotonically increasing.*
4. *The number of tokens in r^- is monotonically decreasing.*
5. *If there is a token in place p_{q_i} and there are d tokens in r^{ctrl} only the transition $t_{q_iq_j}$ is enabled.*
6. *There can only be tokens in r if there are no tokens in r^-.*

The next lemma shows that the net works as intended. The result follows from the previous proposition.

Lemma 8. *Let there be a token in p_{q_i}, $d' \leq d$ tokens in r^-, $d - d'$ tokens in r^{ctrl}, and no tokens in r. Let σ be the minimal length firing sequence such that there is a token in p_{q_j}. Then, after executing σ there are $\max\{0, d' - u\}$ tokens in r^-, $\min\{d, d - d' + u\}$ tokens in r^{ctrl}, and $\max\{0, u - d'\}$ tokens in r.*

On the other hand, if there is a token in p_{q_i}, d tokens in r^{ctrl}, zero tokens in r^- and k tokens in r then, after executing σ there are $k+u$ tokens in r, d tokens in r^{ctrl}, and zero tokens in r^-. □