A full operational semantics for Asynchronous Relational Nets

Carlos Gustavo Lopez Pombo — (1, 3)
Ignacio Vissani — (1, 3)
Ionuţ Ţuţu — (2)
José Luiz Fiadeiro — (2)

(1) Universidad de Buenos Aires, Argentina
(2) Royal Holloway, University of London
(3) Consejo Nacional de Investigaciones Científicas y Tecnológicas (CONICET)

Intro (The Motivation)

The general context: Service-Oriented Computing

- In **Service-Oriented Computing** (SOC), the structure of software systems is intrinsically dynamic since: **a)** they run over globally available computational capabilities and network infrastructure, and **b)** they may require these computational capabilities in the form of services that are procured at run-time to fulfil a given business goal
- The discovery and binding of services is done at run-time by a dedicated middleware which is transparent from the perspective of the executing software artefact

Intro (The Motivation)

A **full operational semantics** for Asynchronous Relational Nets

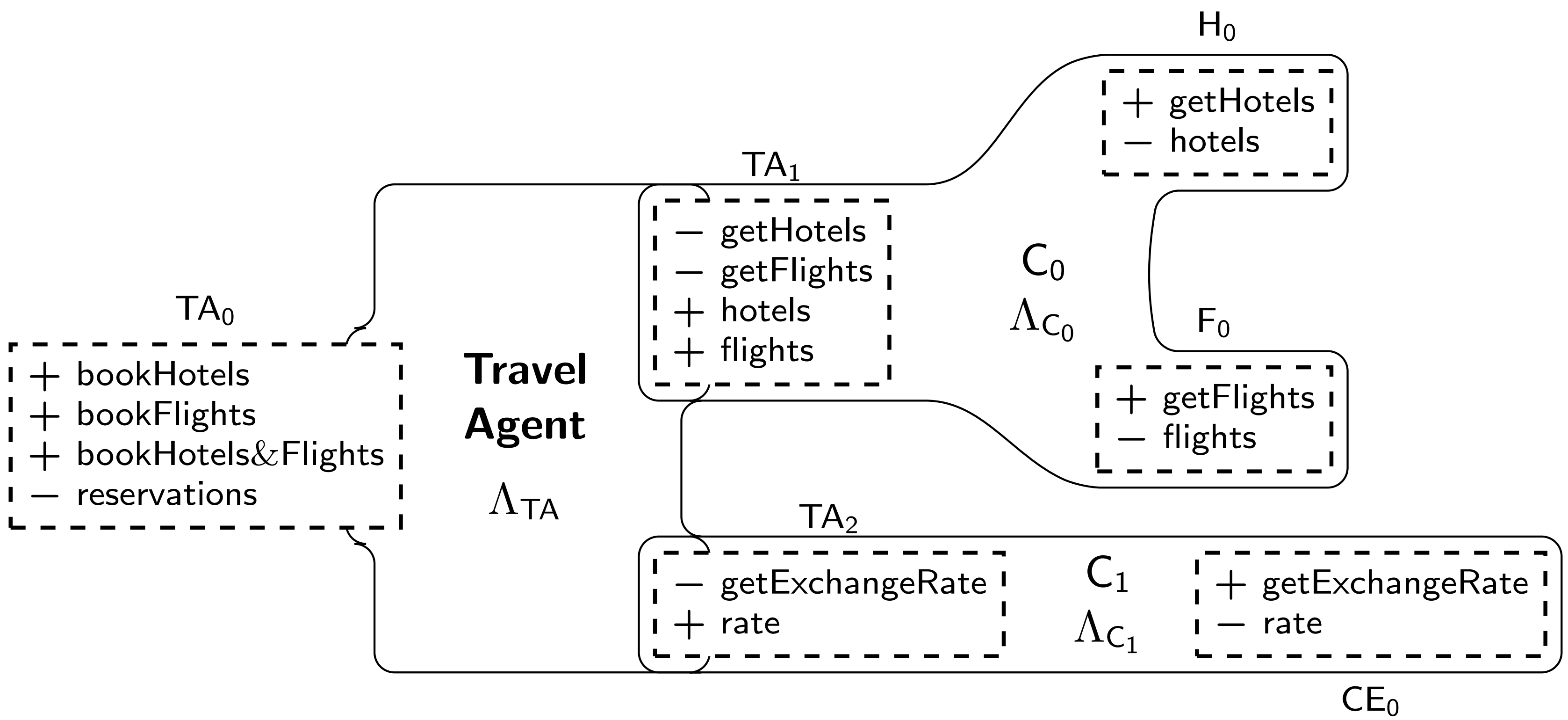
* Properly understanding the behaviour associated to formal models requires to fix meaning to syntactic constructions (i.e., **semantics**)

* The dynamic nature of SOC suggests the definition of a semantics as close as possible to the actions occurring along an execution (i.e., **operational semantics**)

* Actions taking place in an execution define the expected behaviour of the components intervening in it, like the middleware; so we chose to avoid any denotational descriptions (i.e., **full**)

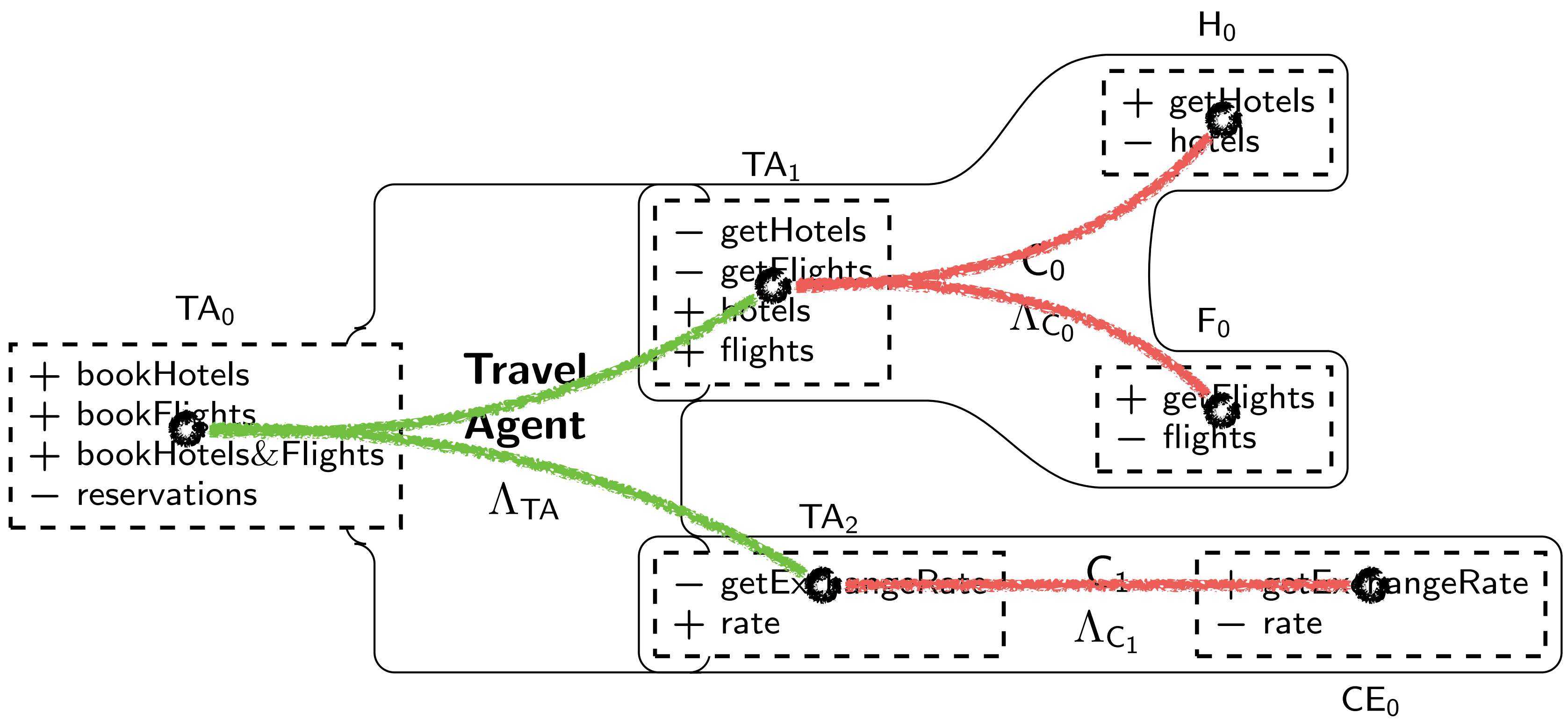
Intro (The Motivation)

A full operational semantics for **Asynchronous Relational Nets**



Intro (The Motivation)

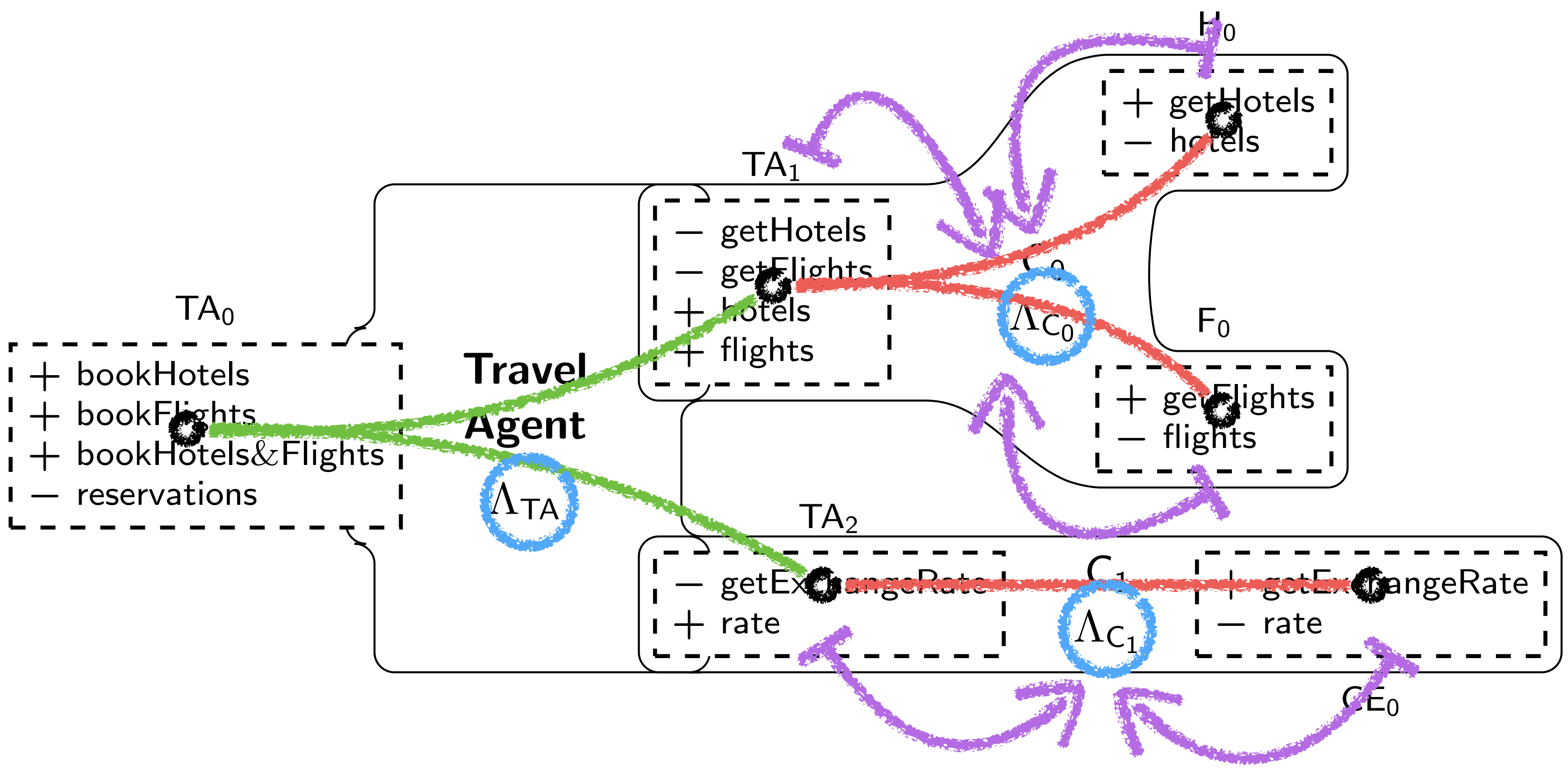
A full operational semantics for **Asynchronous Relational Nets**



An ARN is a hypergraph-based structure whose nodes are the **ports**, and has two types of hyperedges: **communication channels** and **processes**

Intro (The Motivation)

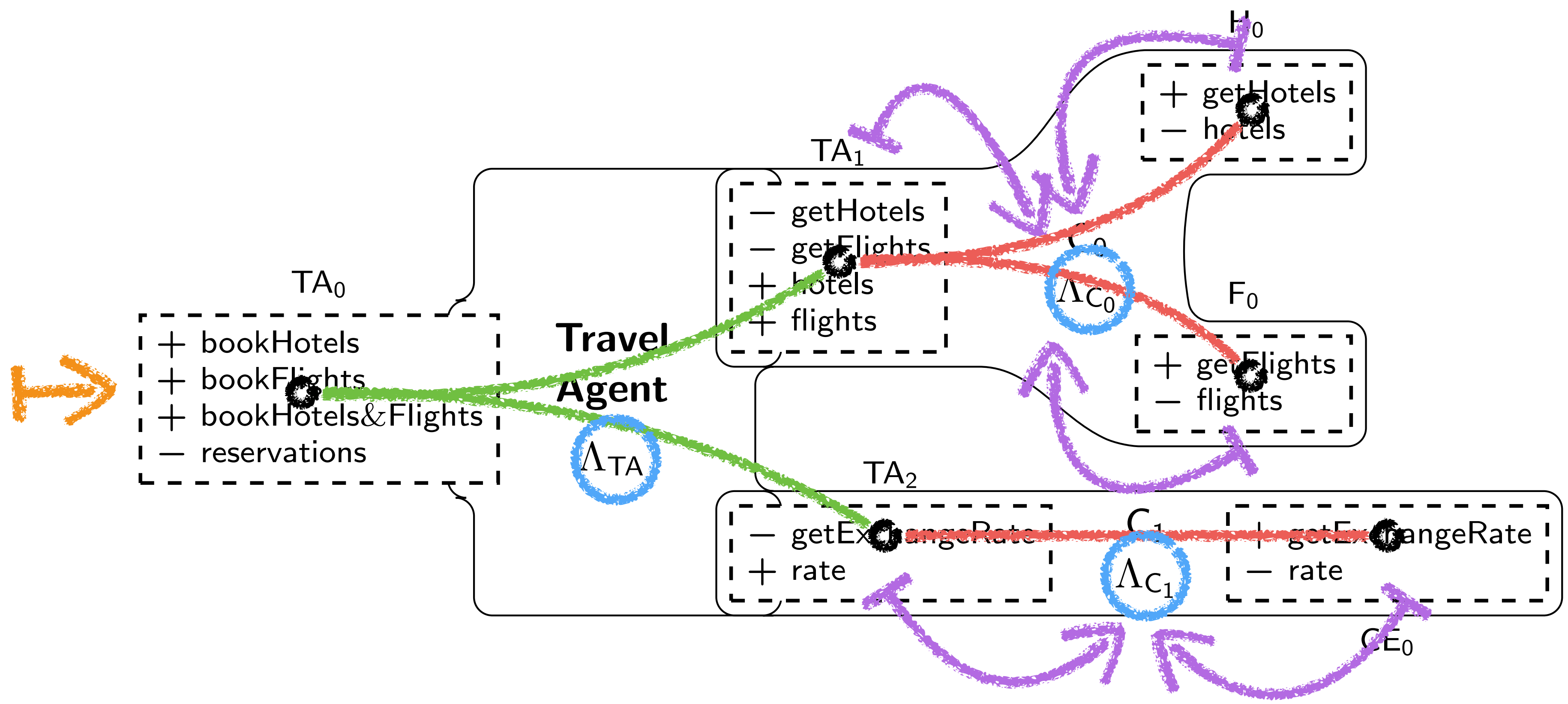
A full operational semantics for **Asynchronous Relational Nets**



Each edge is labeled with a Müller automaton, in the case of **processes** on the language of the **ports**, in the case of **communication channels** on a new language to which the language of the **ports** are **mapped by injections**,

Intro (The Motivation)

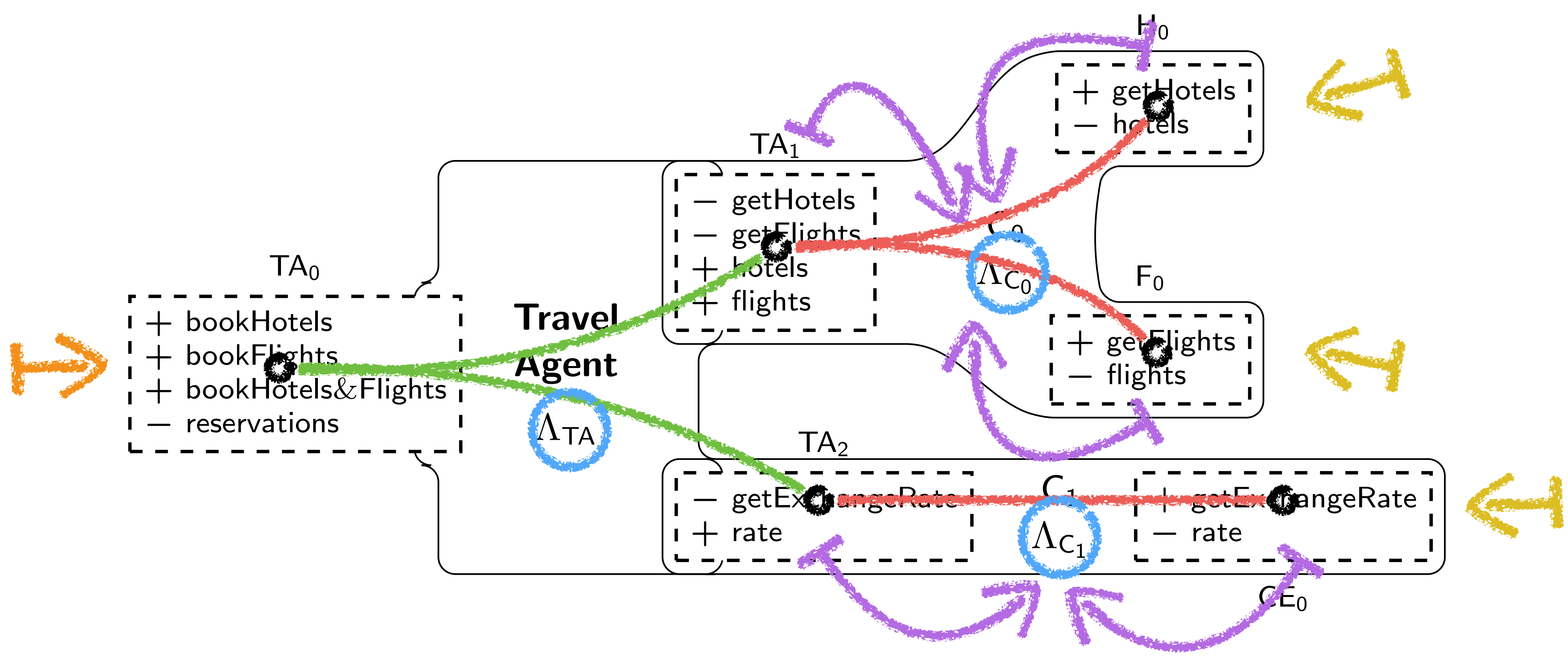
A full operational semantics for **Asynchronous Relational Nets**



Nodes that are only incident to **processes** are called **provides points**,

Intro (The Motivation)

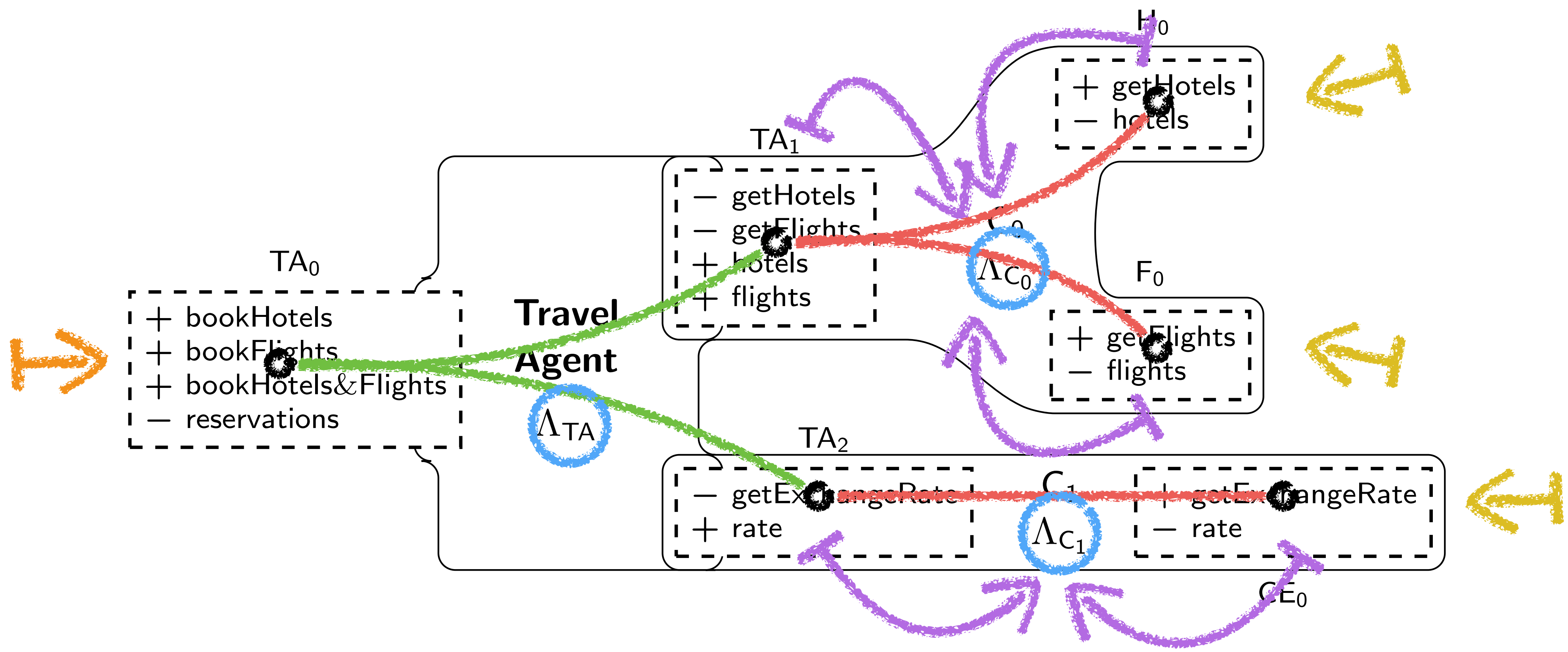
A full operational semantics for **Asynchronous Relational Nets**



Nodes that are only incident to **processes** are called **provides points**,
while those that are only incident to **communication channels** are called **require points**

Intro (The Motivation)

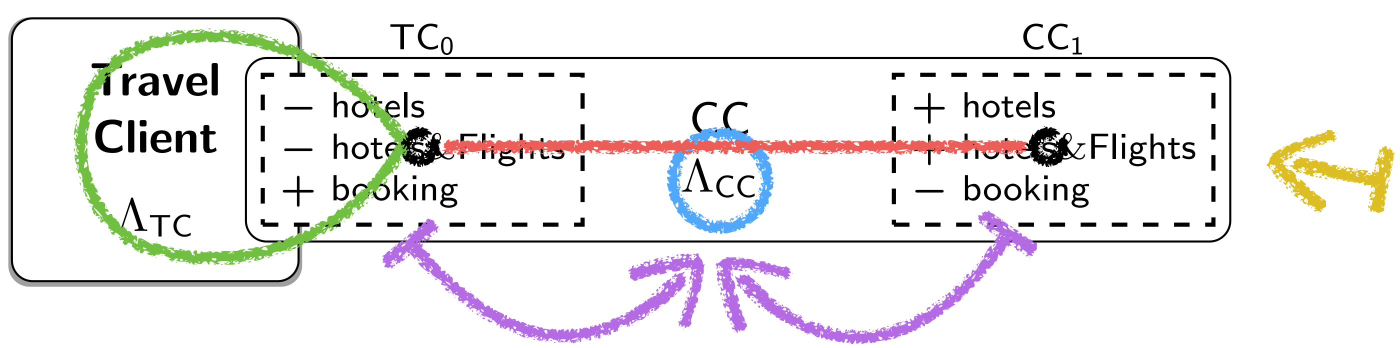
A full operational semantics for **Asynchronous Relational Nets**



If an ARN has **provides points**, it is said to be a **service** as it can be invoked through them,

Intro (The Motivation)

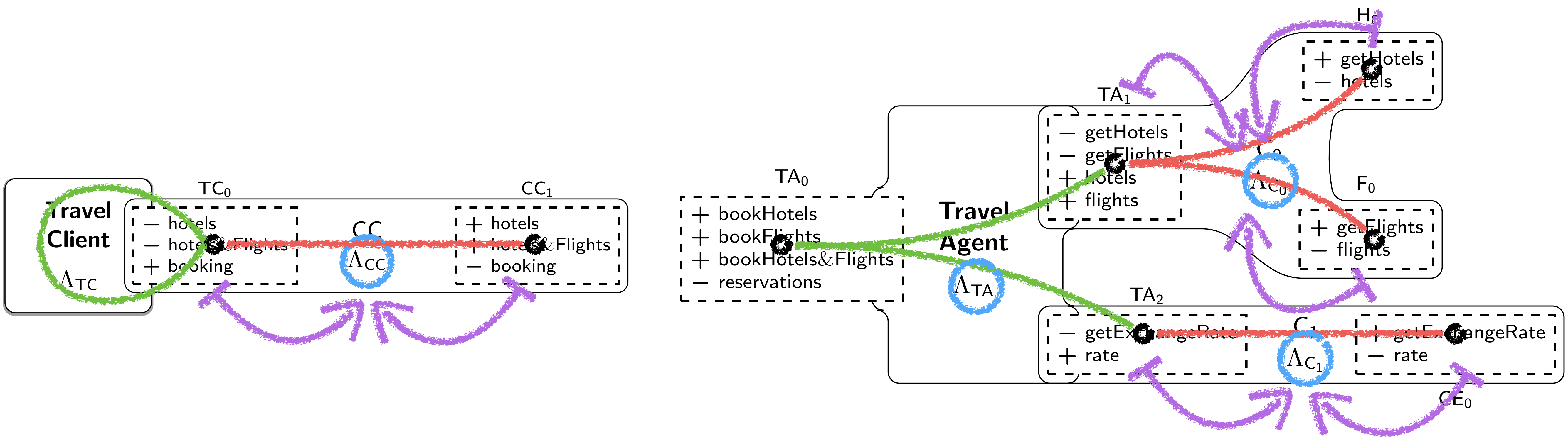
A full operational semantics for **Asynchronous Relational Nets**



If an ARN has **provides points**, it is said to be a **service** as it can be invoked through them,
while if it only have **requires points**, it is said to be an **activity**, meaning that it can not be invoked.

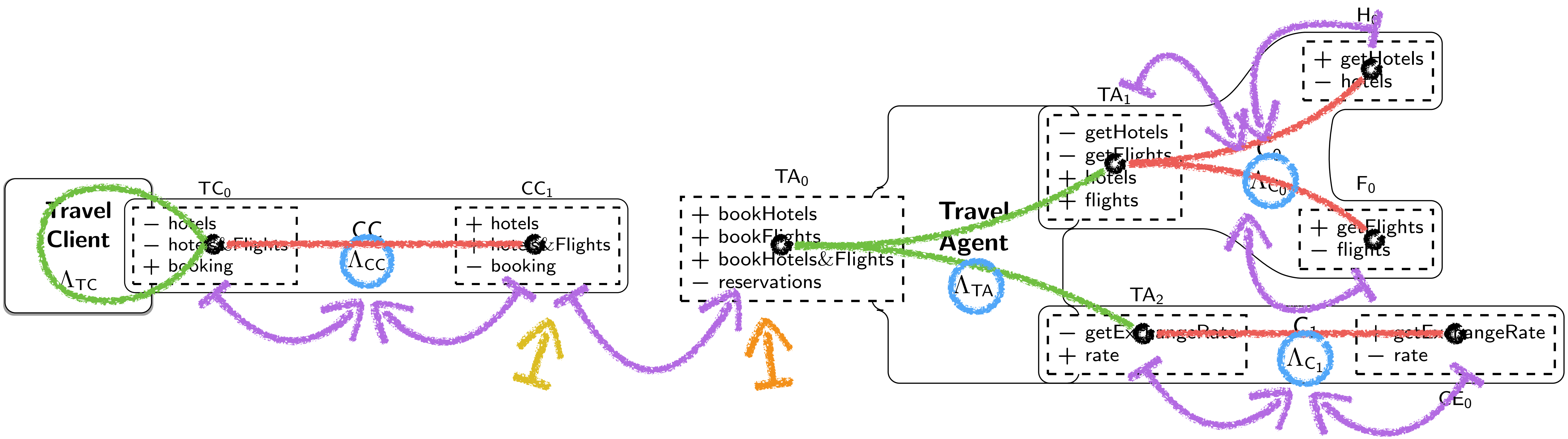
Intro (The Motivation)

A full operational semantics for **Asynchronous Relational Nets**



Intro (The Motivation)

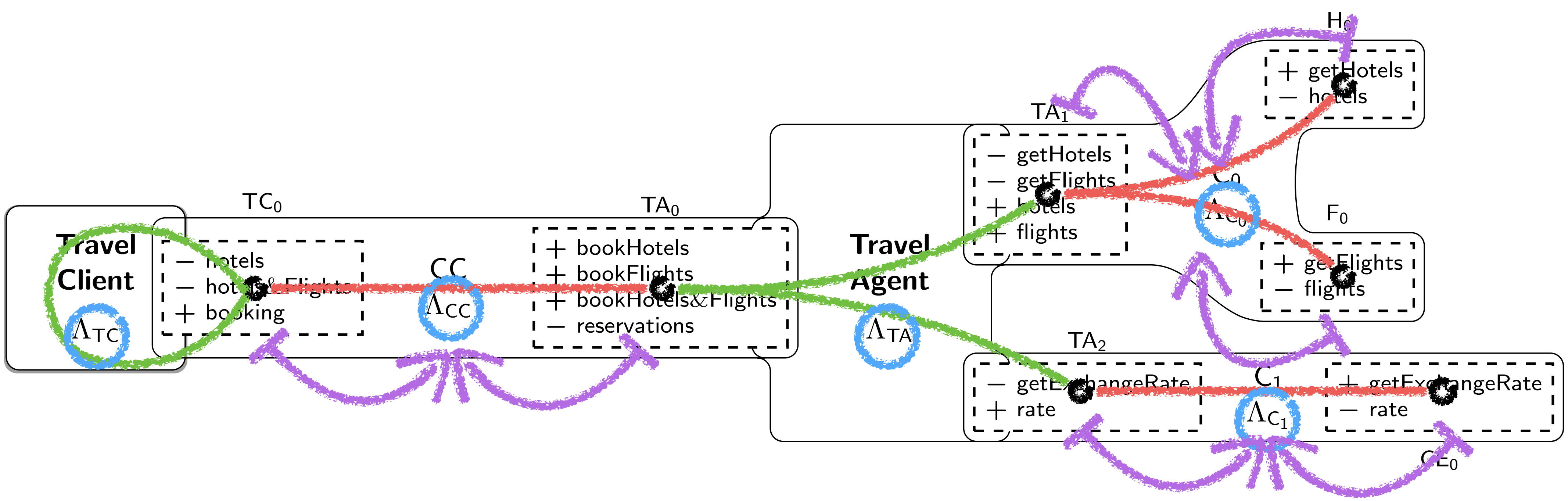
A full operational semantics for **Asynchronous Relational Nets**



The composition of an **activity** with a **service** is done by **injectively mapping** the language of a **requires points** of an activity to the language of a **provides point** of a service.

Intro (The Motivation)

A full operational semantics for **Asynchronous Relational Nets**

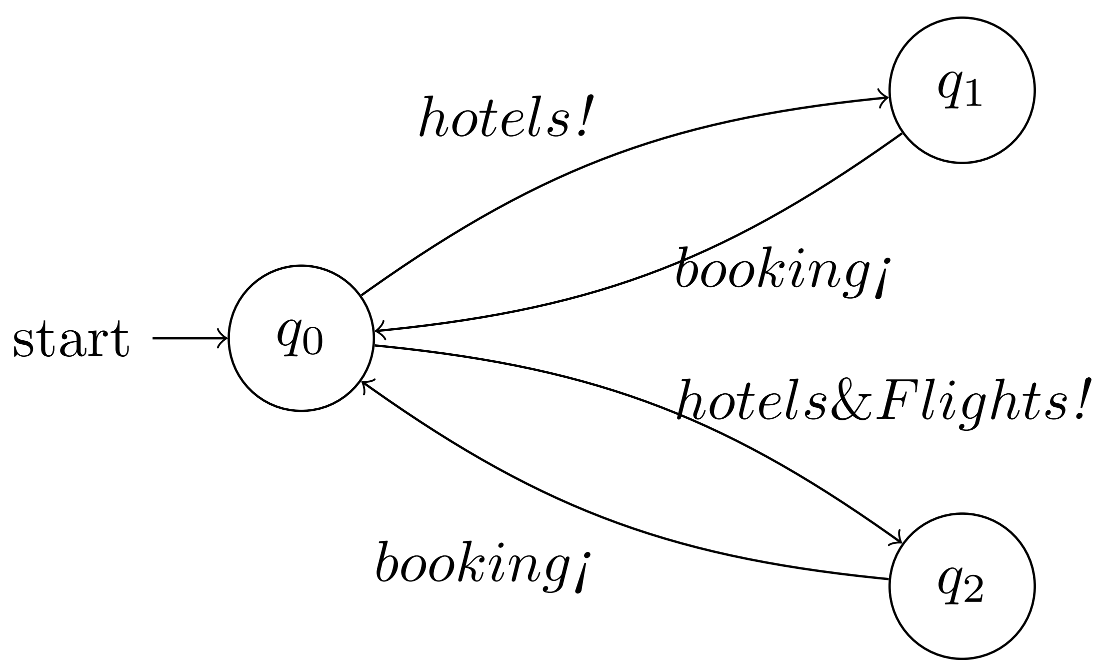


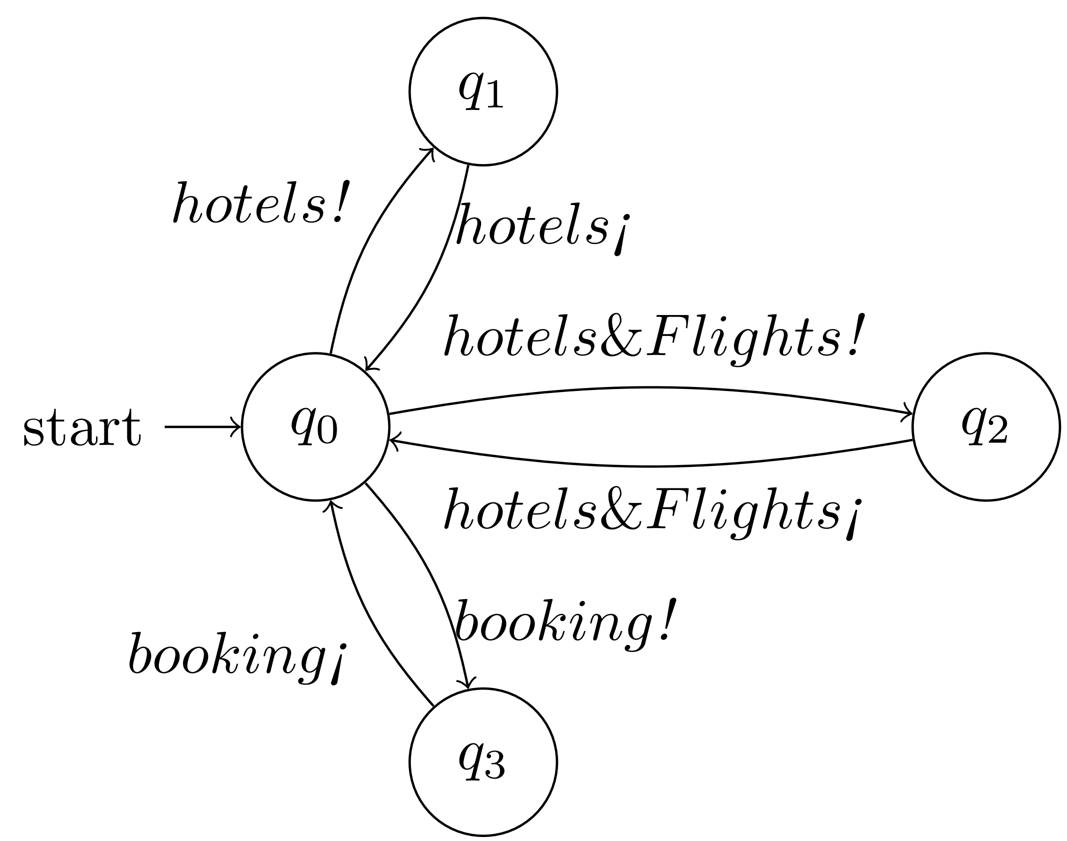
The composition of an **activity** with a **service** is done by **injectively mapping** the language of a **requires points** of an activity to the language of a **provides point** of a service.

Informally speaking

Execution of activities

Internal transitions

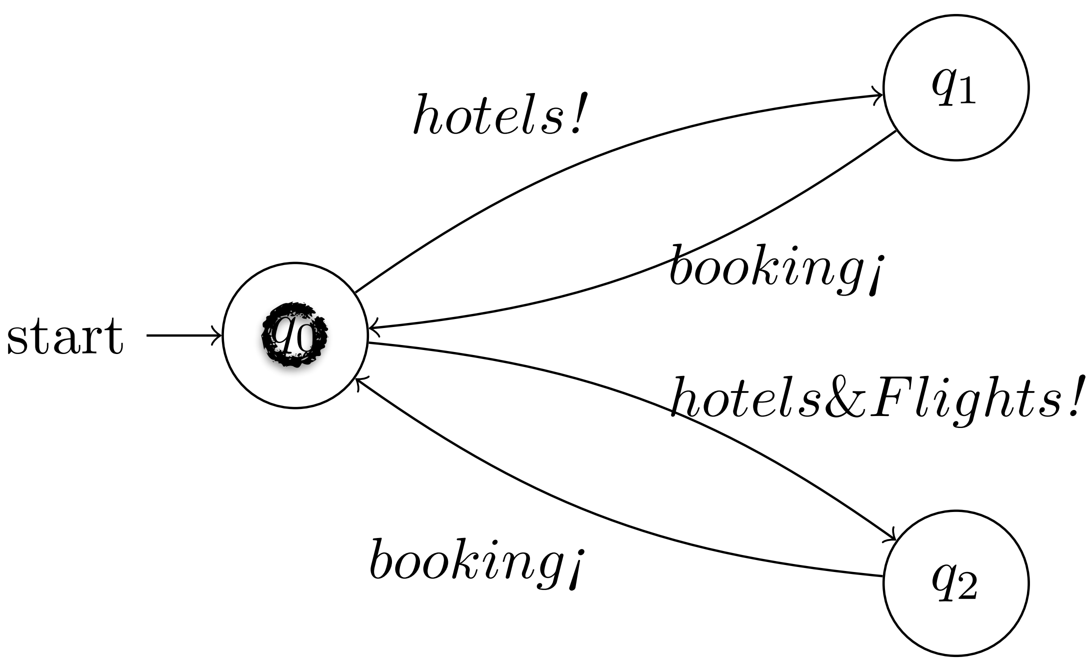


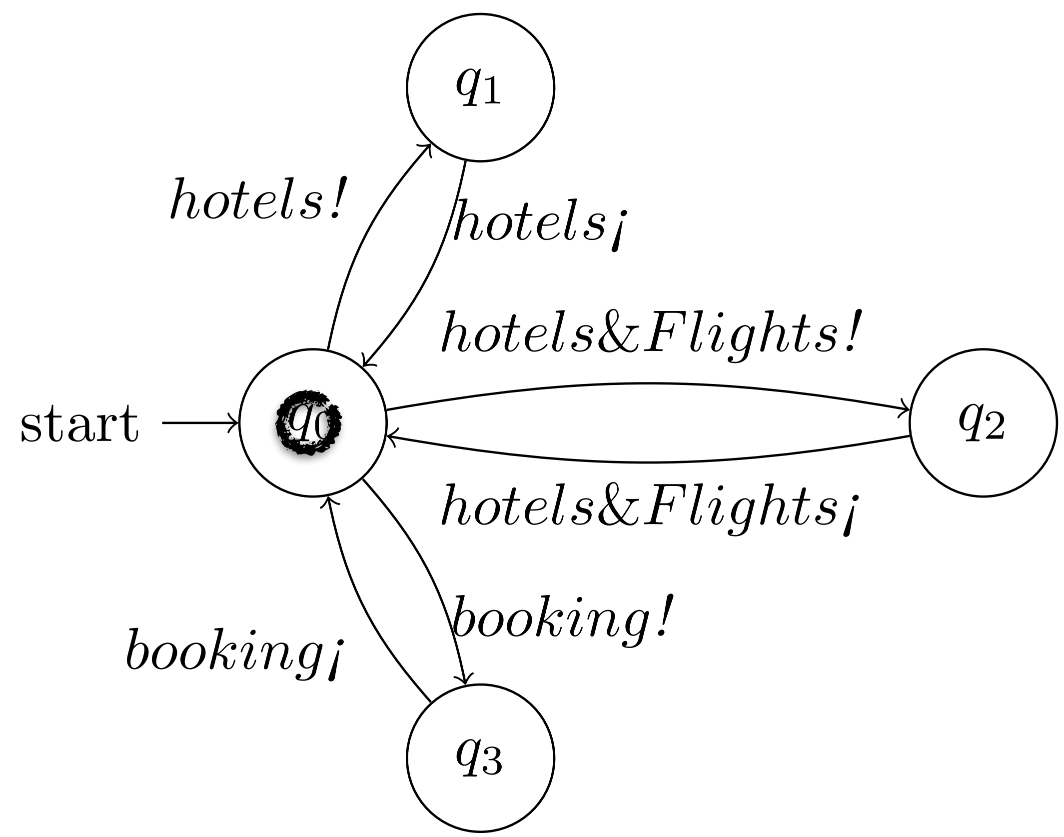


Informally speaking

Execution of activities

Internal transitions



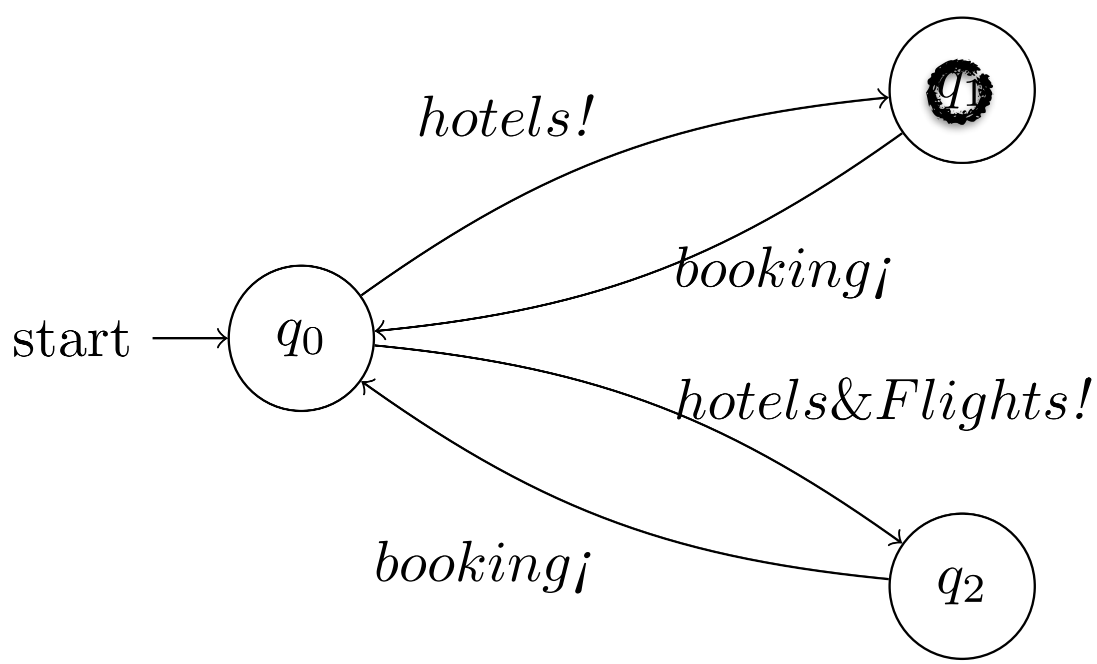


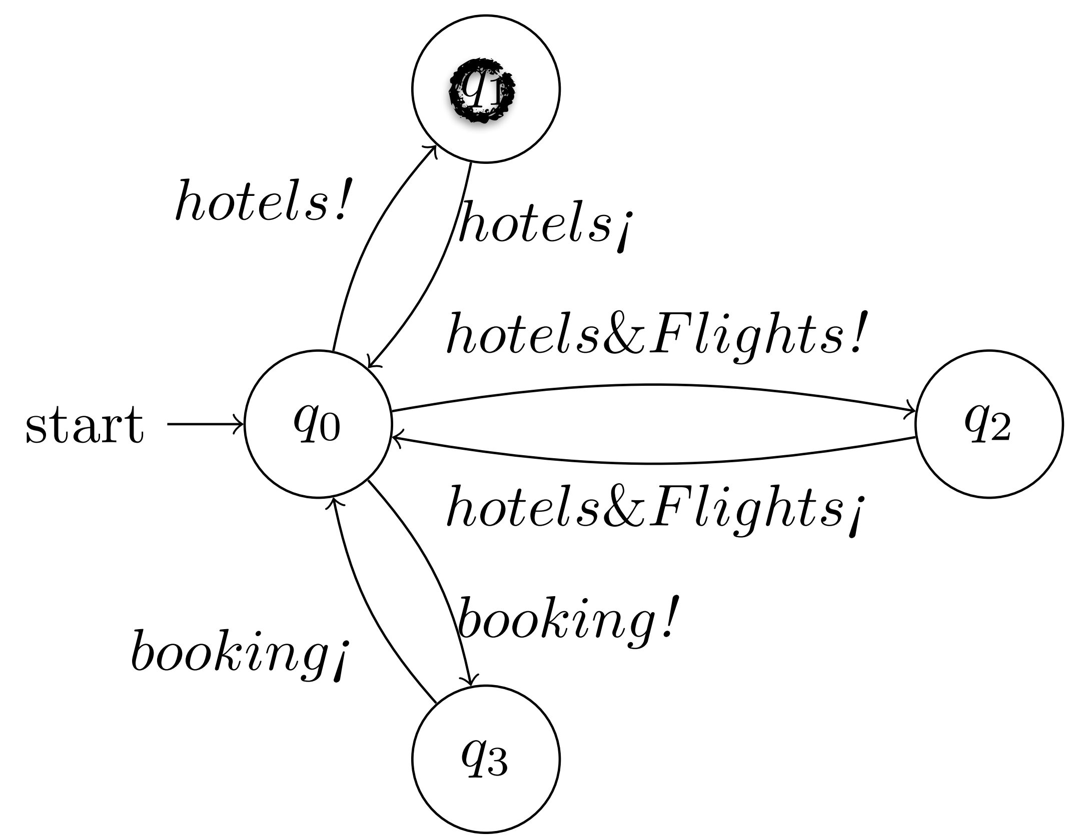
(q0, q0)

Informally speaking

Execution of activities

Internal transitions



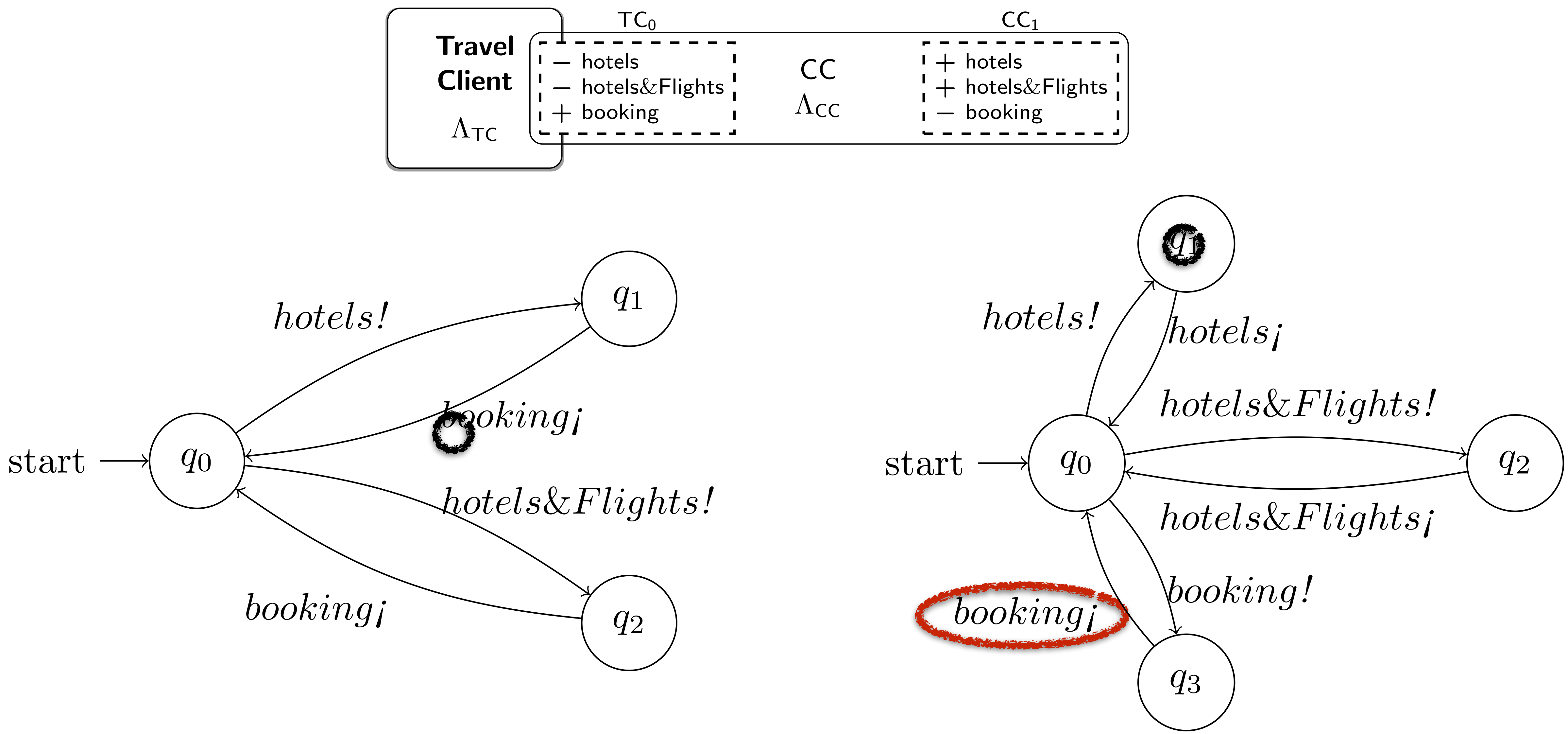


$(q0, q0) \xrightarrow{\text{hotels!}} (q1, q1)$

Informally speaking

Execution of activities

Internal transitions

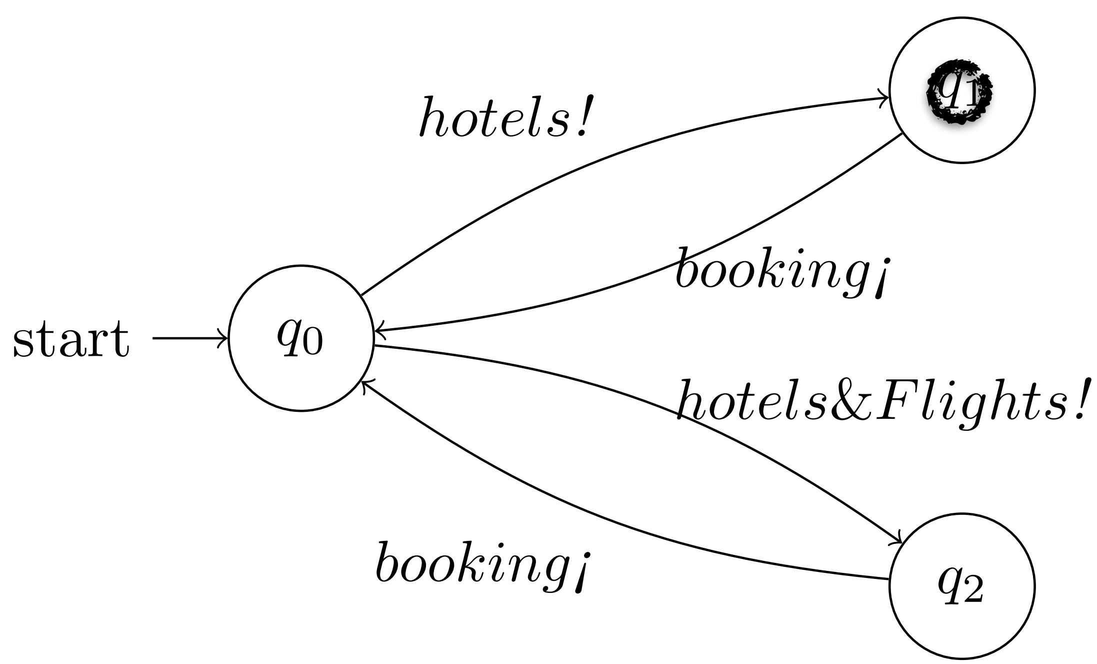


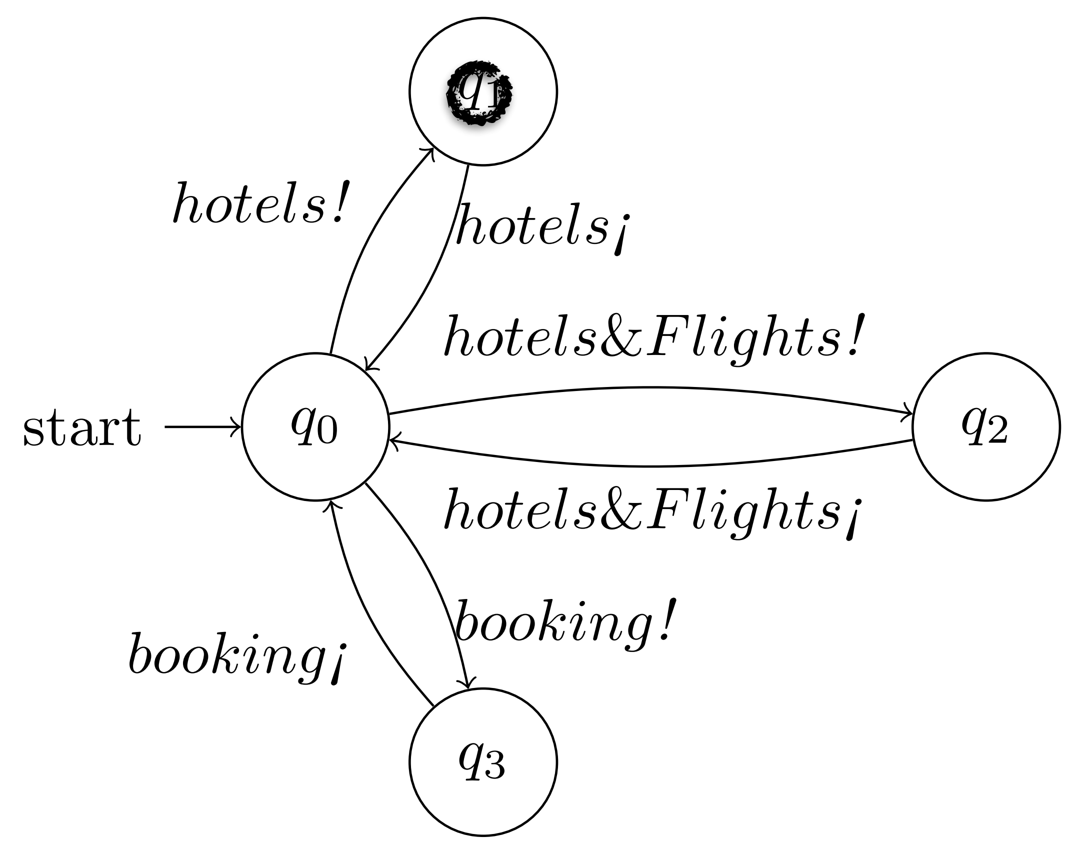
$$(q0,\ q0) \xrightarrow{\text{hotels!}} (q1,\ q1)$$

Informally speaking

Execution of activities

Internal transitions



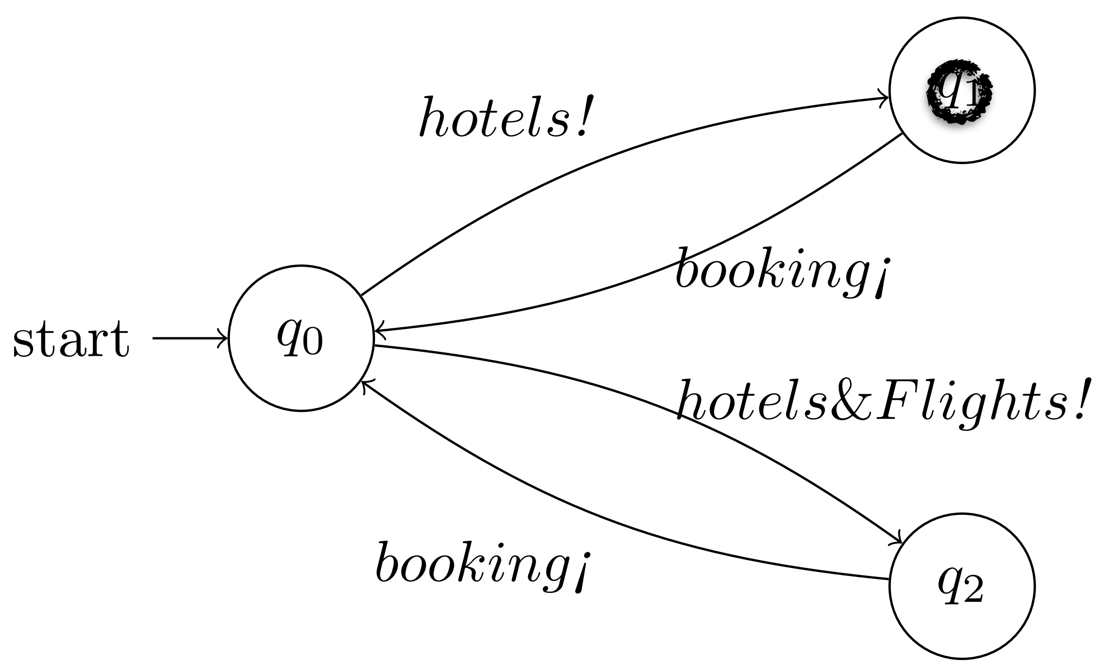


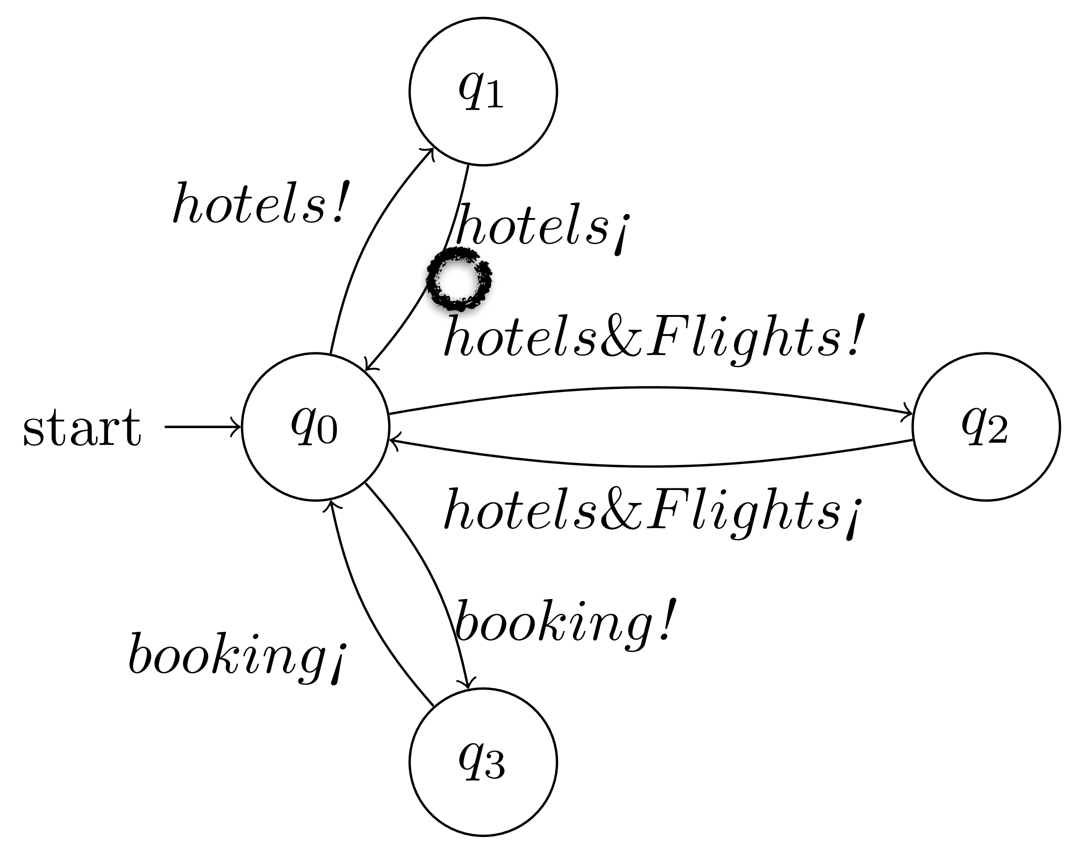
$(q0, q0) \xrightarrow{\text{hotels!}} (q1, q1)$

Informally speaking

Execution of activities

Internal transitions



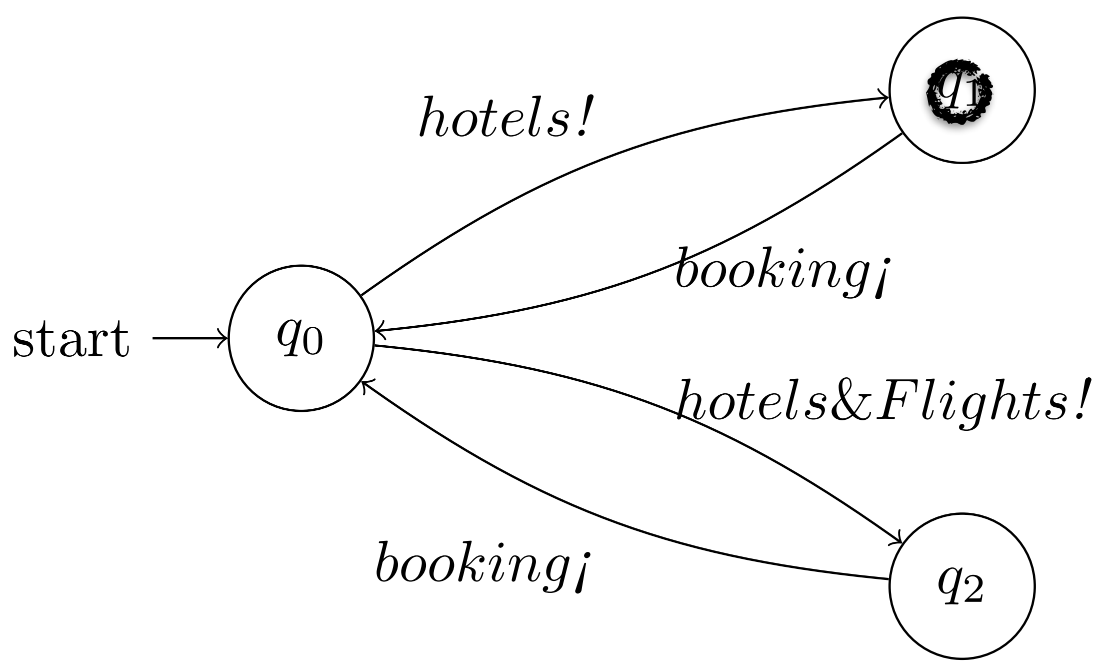


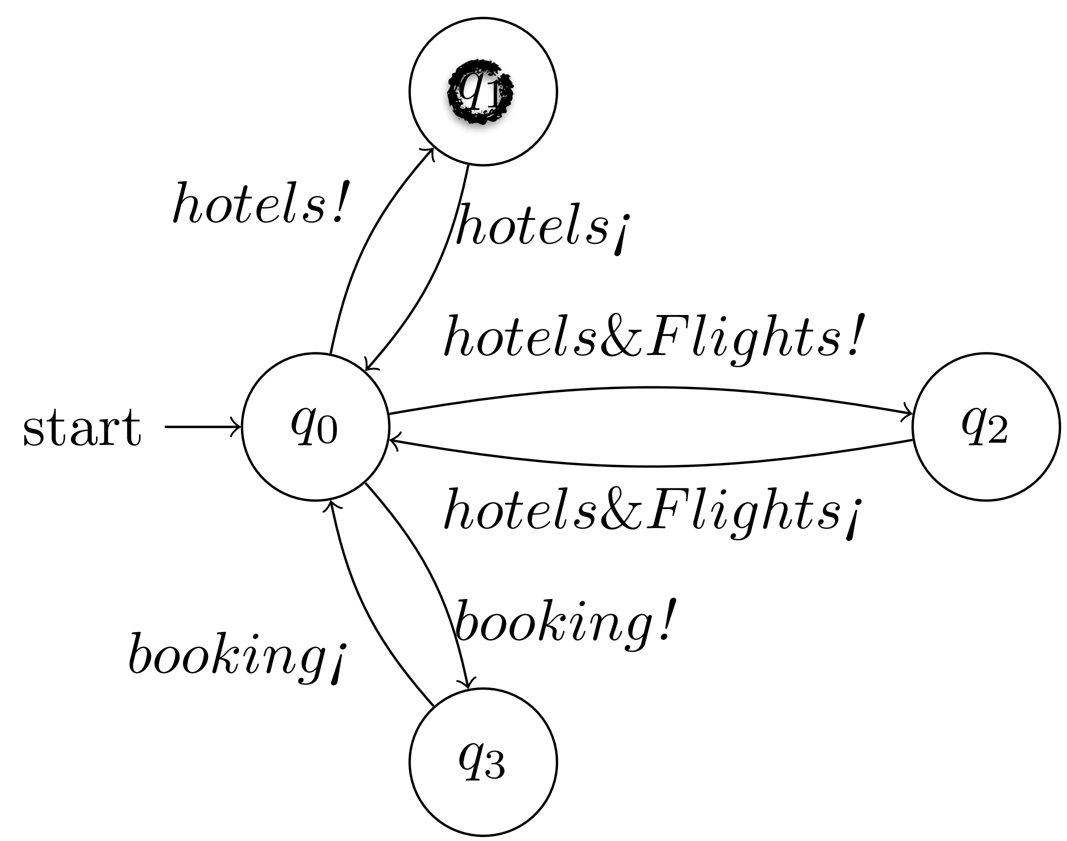
$$(q0, q0) \xrightarrow{\text{hotels!}} (q1, q1)$$

Informally speaking

Execution of activities

Internal transitions



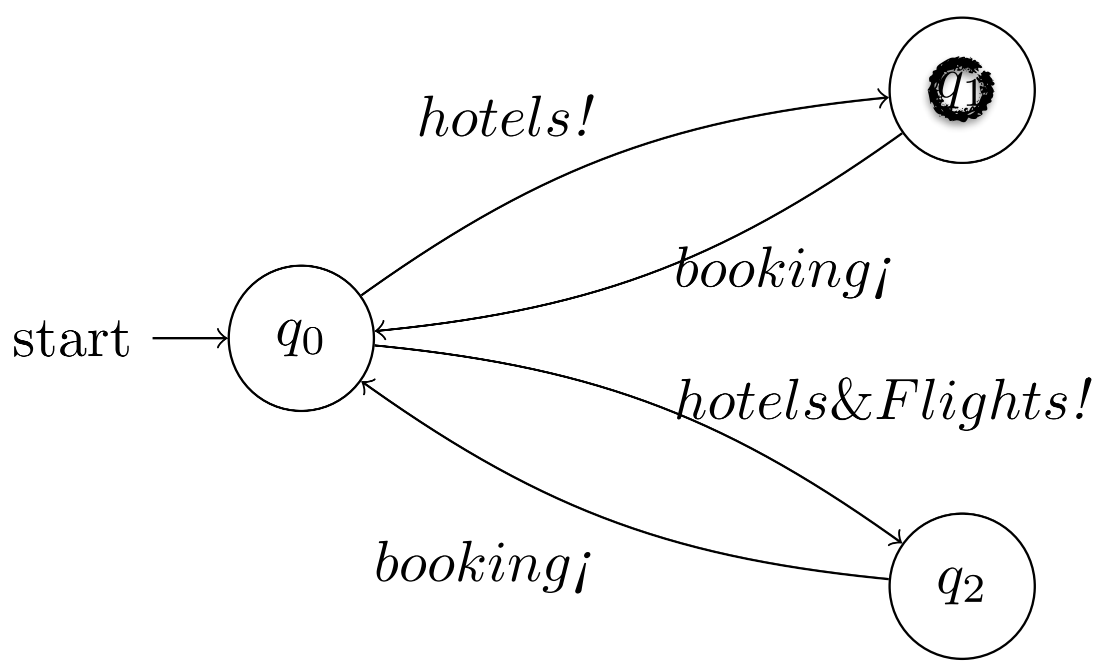


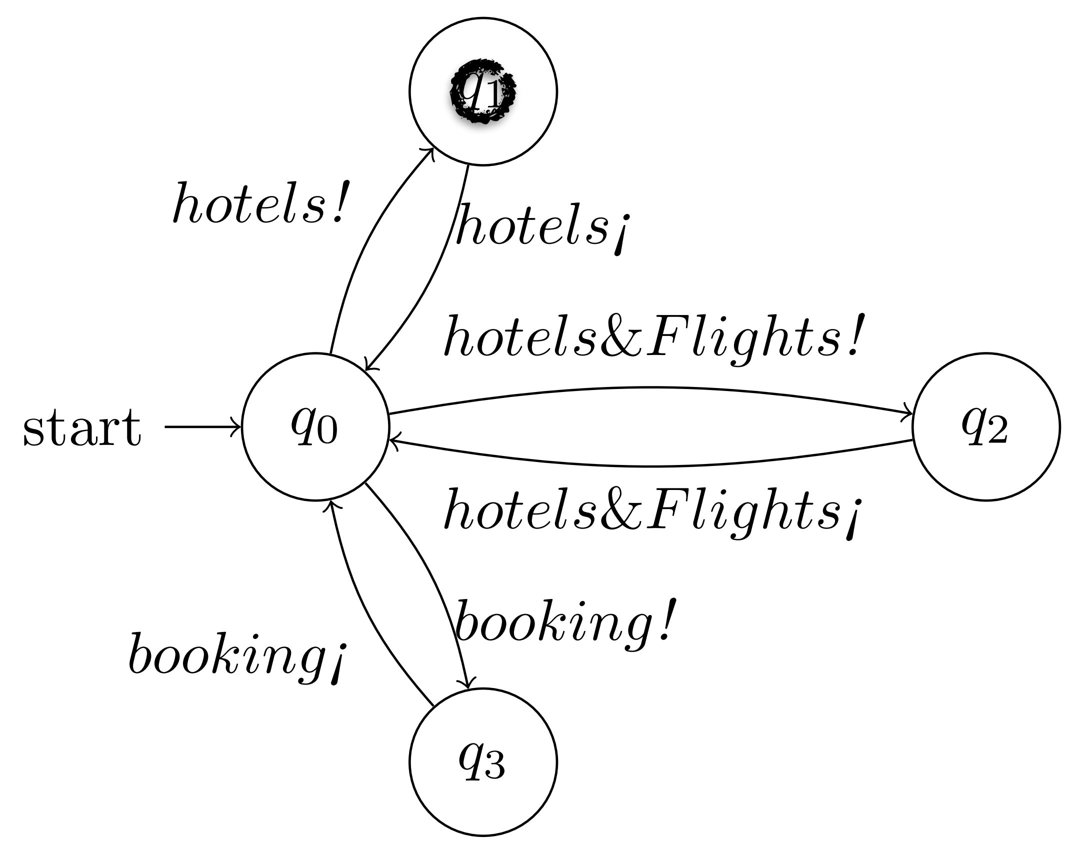
$$(q0, q0) \xrightarrow{\text{hotels!}} (q1, q1)$$

Informally speaking

Execution of activities

Reconfiguration actions



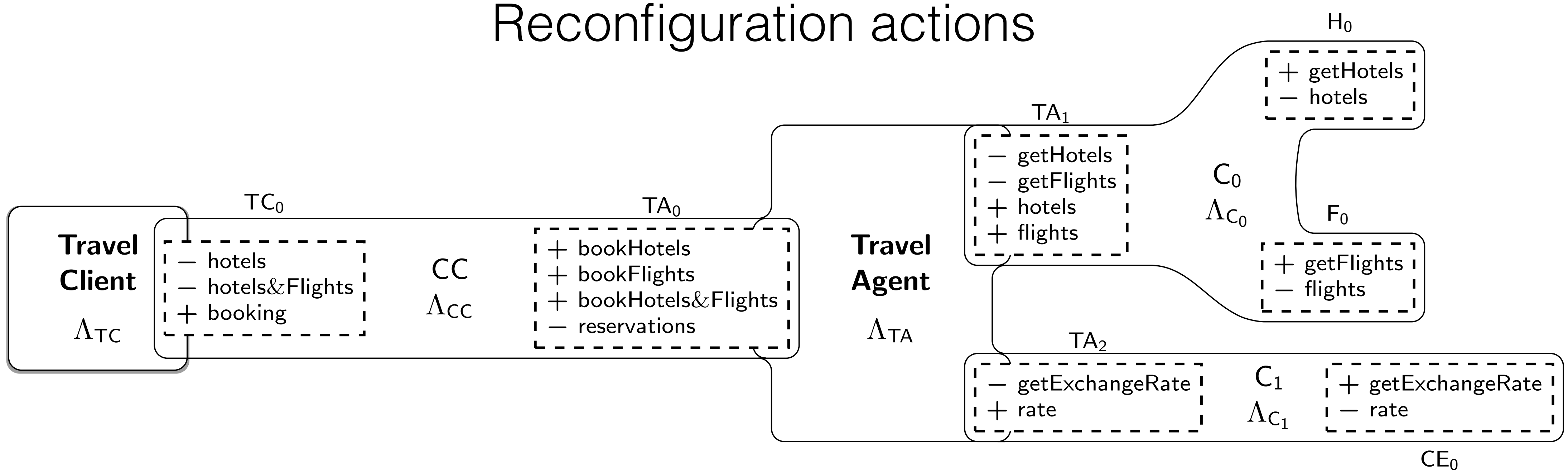


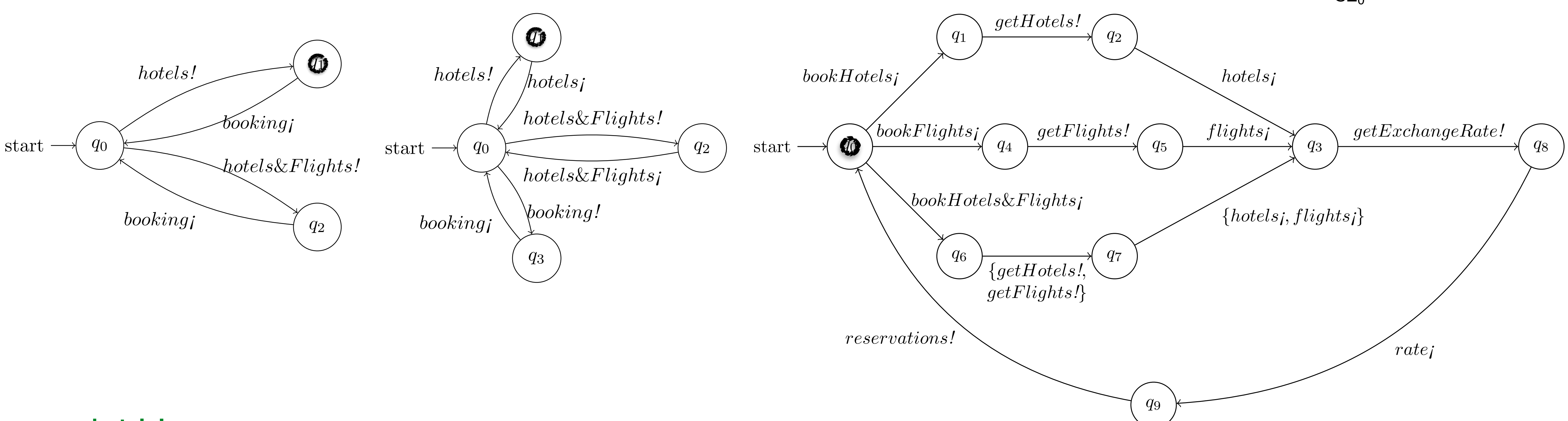
... $\xrightarrow{\text{hotels!}}$ (q1, q1)

Informally speaking

Execution of activities

Reconfiguration actions



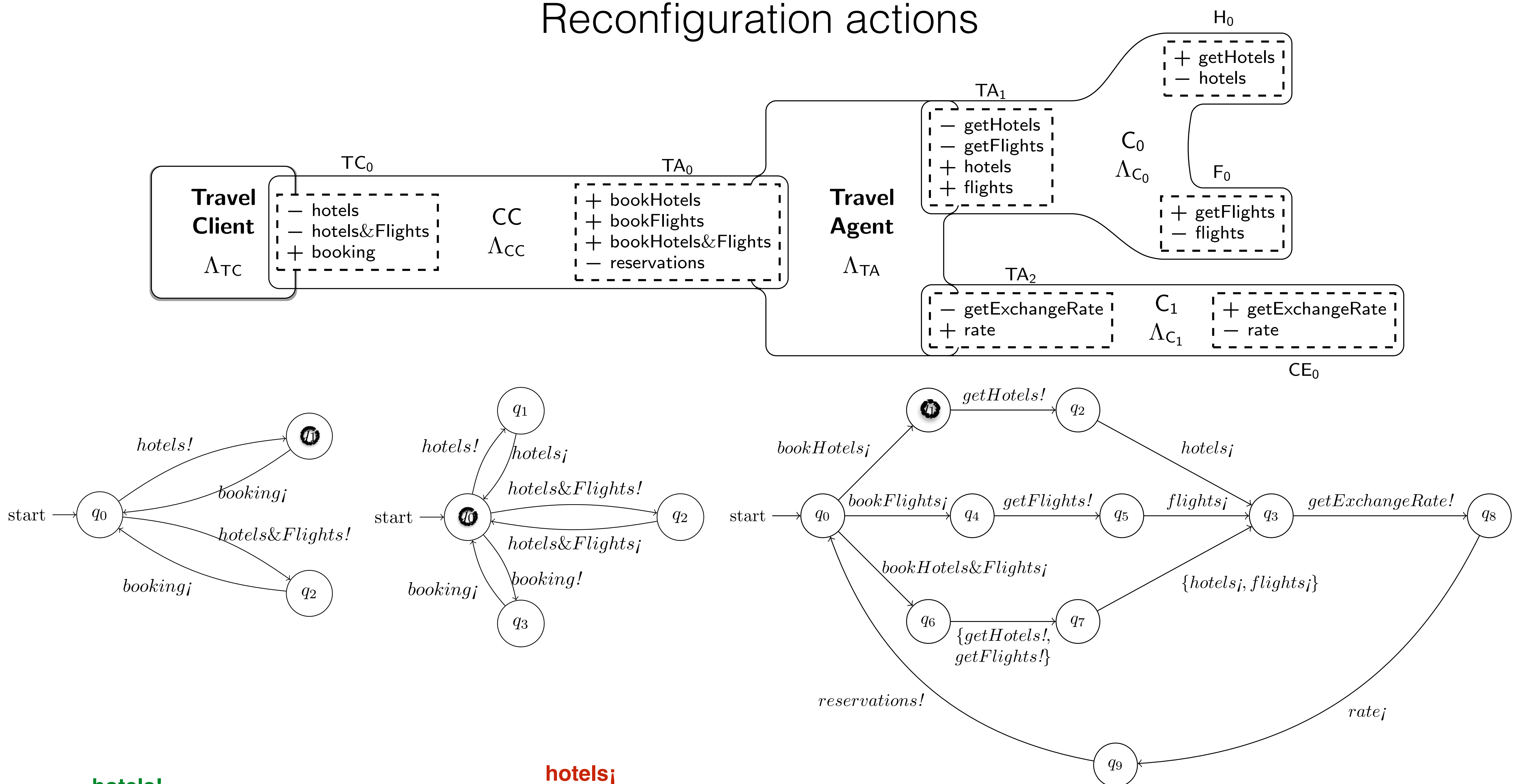


... $\xrightarrow{\text{hotels!}}$ (q1, q1, q0)

Informally speaking

Execution of activities

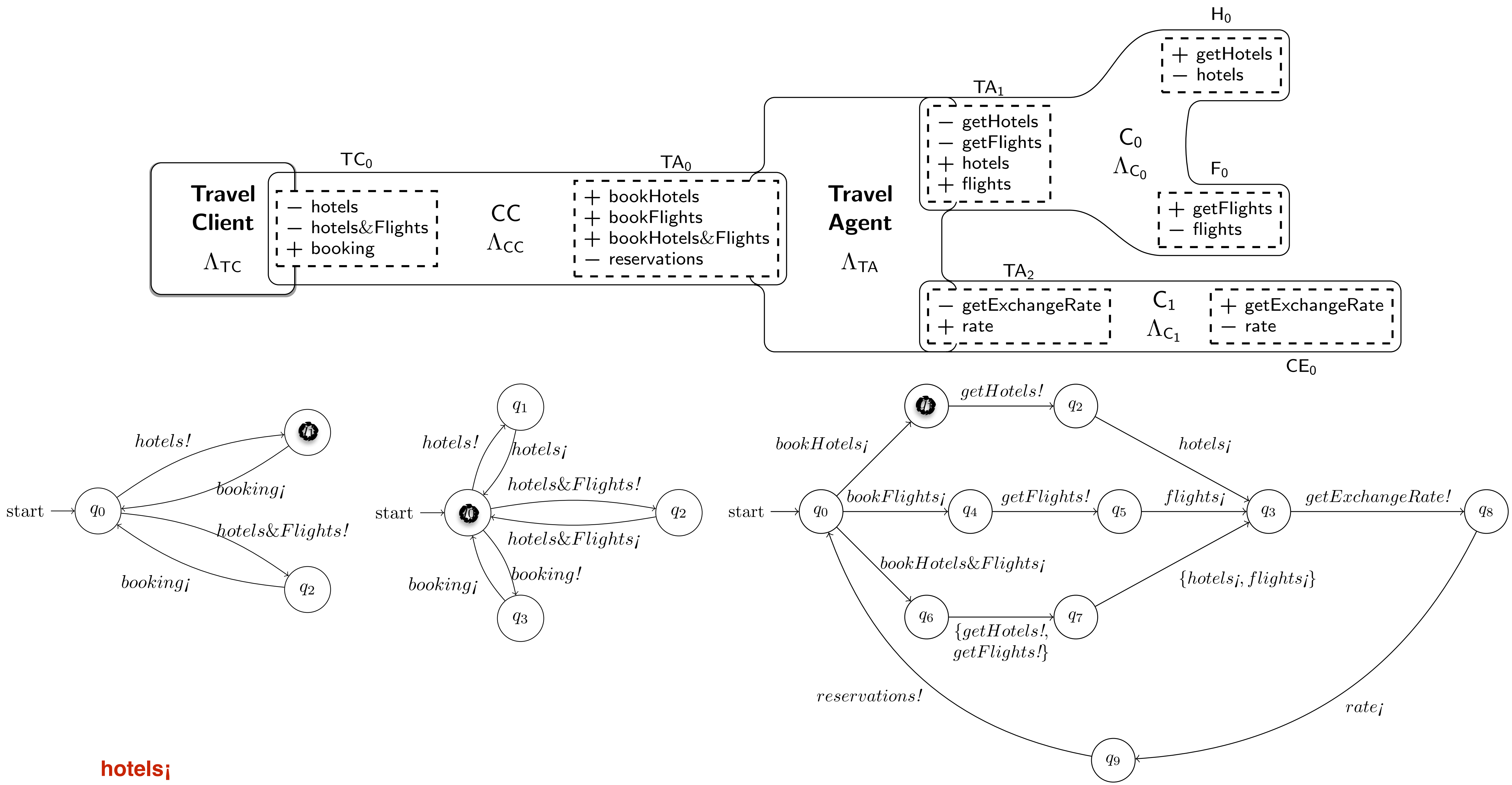
Reconfiguration actions



... $\xrightarrow{\text{hotels!}}$ (q1, q1, q0) $\xrightarrow{\text{hotels¡ bookHotels¡}}$ (q1, q0, q1)

Informally speaking

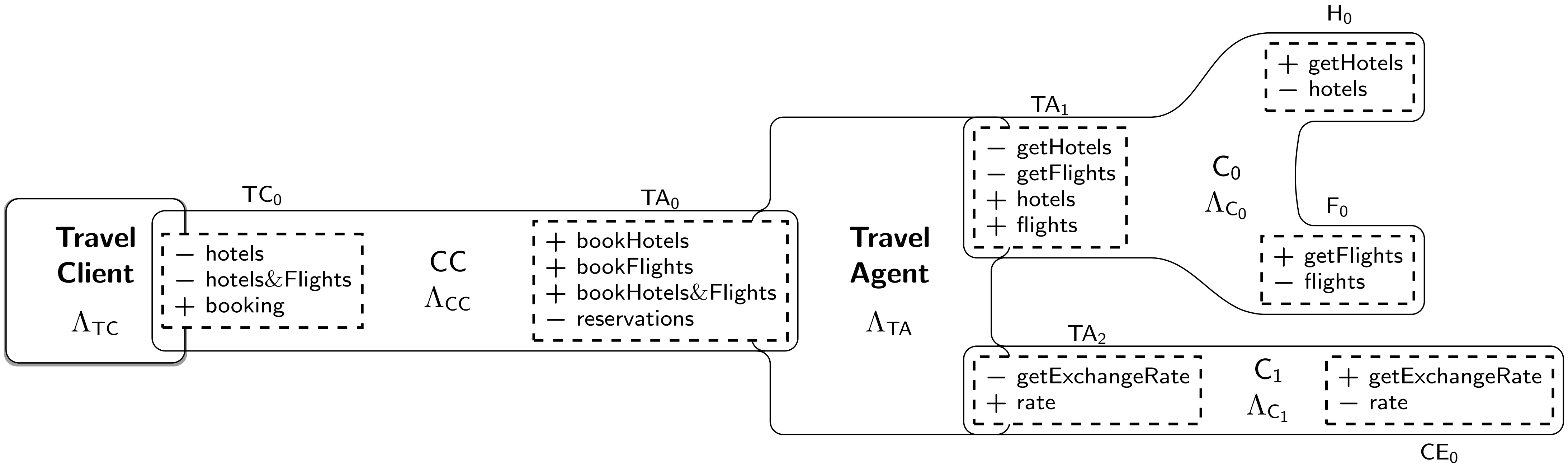
Execution of activities

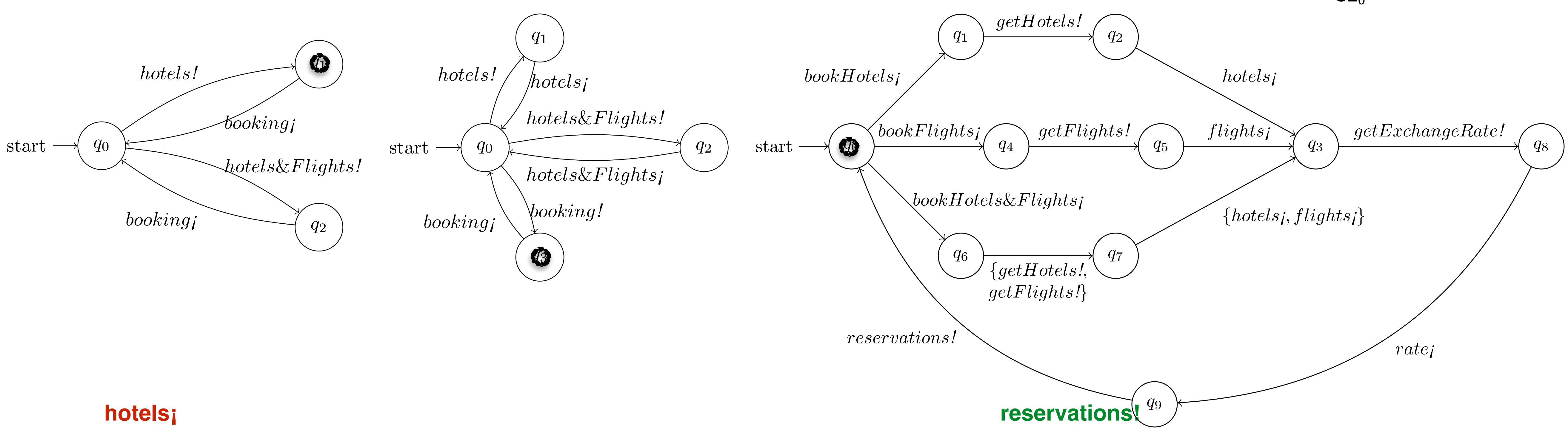


… $\xrightarrow[\text{bookHotels¡}]{\text{hotels¡}}$ (q1, q0, q1)

Informally speaking

Execution of activities

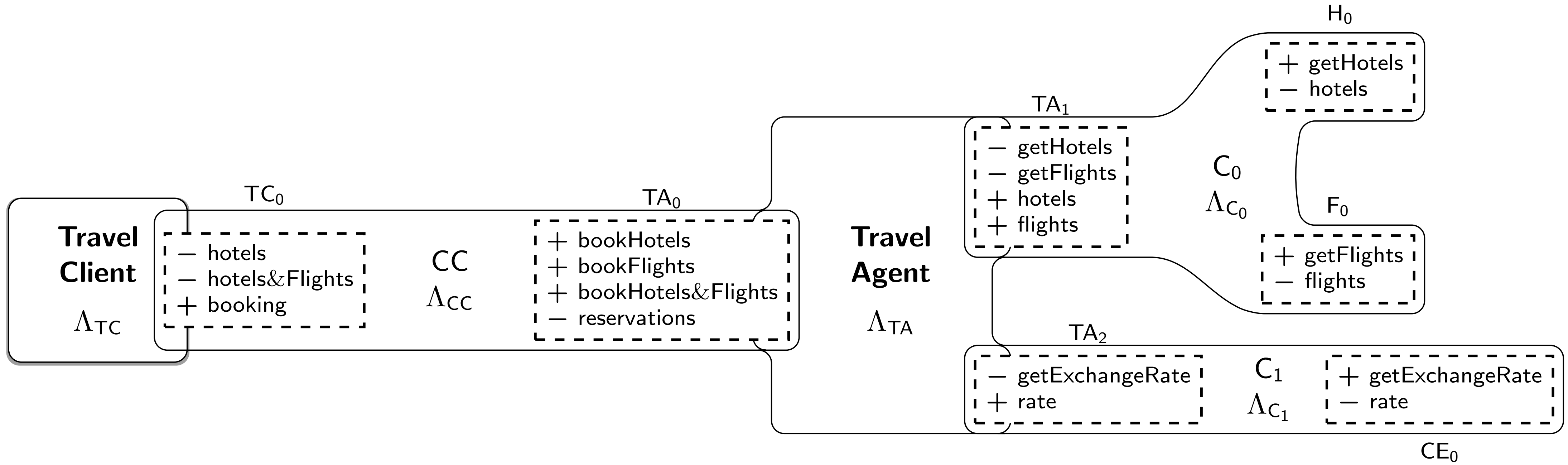


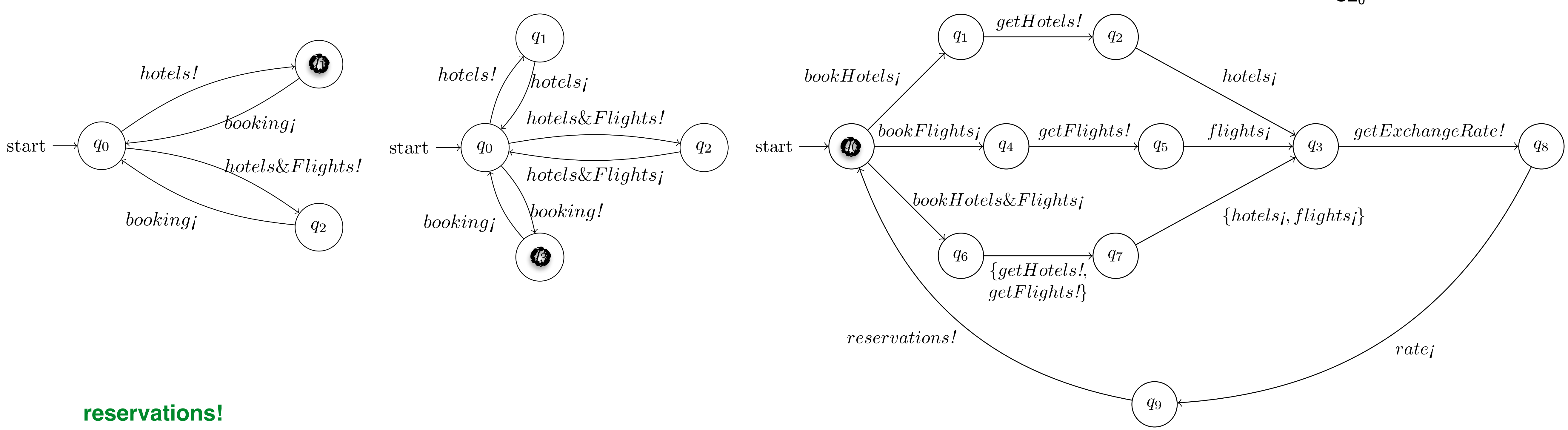


... $\xrightarrow{\text{hotels¡, bookHotels¡}}$ (q1, q0, q1) ... ... (q1, q0, q9) $\xrightarrow{\text{reservations!, booking!}}$ (q1, q3, q0)

Informally speaking

Execution of activities

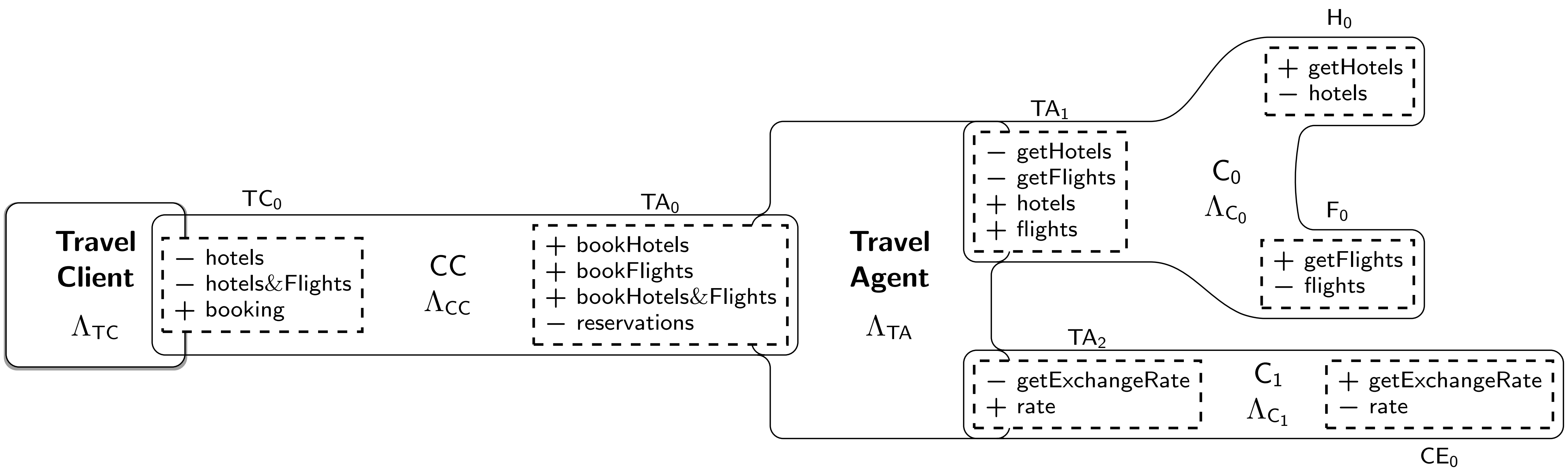


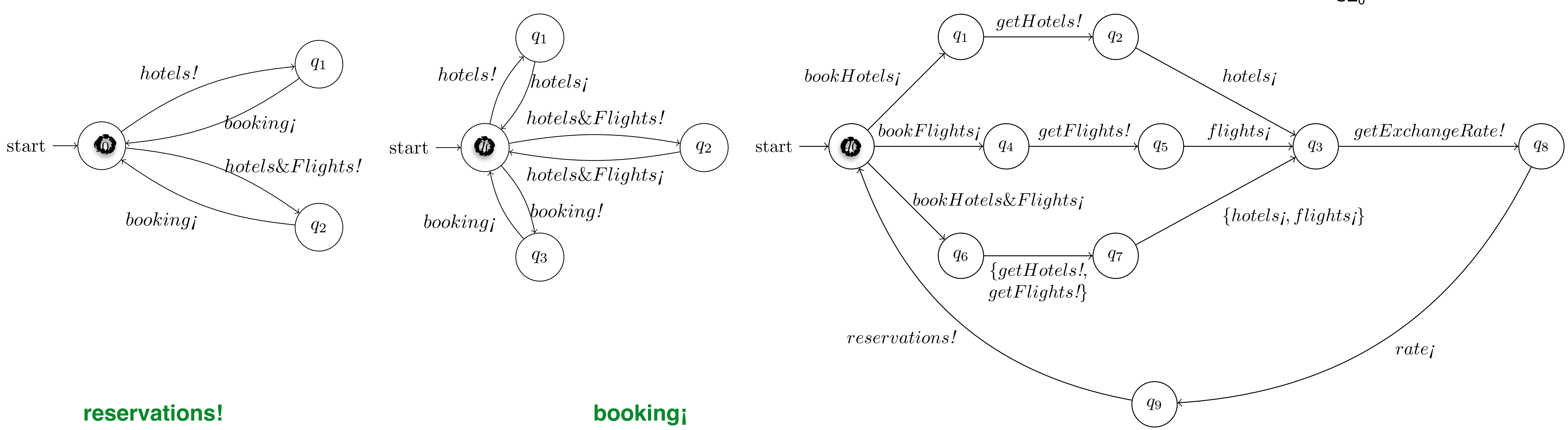


... reservations! booking! → (q1, q3, q0)

Informally speaking

Execution of activities





… $\xrightarrow{\text{reservations! booking!}}$ (q1, q3, q0) $\xrightarrow{\text{booking¡ booking¡}}$ (q0, q0, q0)

Formally speaking

Execution of activities

Transition system

- A **repository** is a family $\{A_i\}_{i \text{ in } I}$ such that, for all i in I, A_i is a service
- A **state** of an activity A is $\{q_i\}_{i \text{ in PUC}}$, a family of states, one for each of the automaton of A
- The **transition system** of an activity A in a repository Rep is a structure (S, —>) where:
  - S = { (A, q) | q is a state of A }, and
  - (A, q) —> (A', q') iff

$\xrightarrow{I}$ A = A' and q $\xrightarrow{I}$ q' in A, or

$\xrightarrow[\text{Rep}]{I}$ A' = A $\underset{\{m_j\}_{j \text{ in } J}}{+}$ $\{B_j\}_{j \text{ in } J}$, $\{B_j\}$ in Rep and q $\xrightarrow{I}$ q' in A'

Formally speaking

Execution of activities

Transition system

- A **repository** is a family $\{A_i\}_{i \text{ in } I}$ such that, for all i in I, A_i is a service
- A **state** of an activity A is $\{q_i\}_{i \text{ in } PUC}$, a family of states, one for each of the automaton of A
- The **transition system** of an activity A in a repository Rep is a structure (S, —>) where:
  - S = { (A, q) | q is a state of A }, and
  - (A, q) —> (A', q') iff

$\xrightarrow{l}$ A = A' and q $\xrightarrow{l}$ q' in A, or

$\xrightarrow[\text{Rep}]{l}$ $A' = A \underset{\{m_j\}_{j \text{ in } J}}{+} \{B_j\}_{j \text{ in } J}$, $\{B_j\}$ in Rep and q $\xrightarrow{l}$ q' in A'

Formally speaking

Execution of activities

Binding of services

- If A is an activity with a requires point r and A' is a service with a provides point p, and m: Mr —> Mp a polarity preserving injective function then,

  A + A' is the activity resulting from glueing the requires point r of A with the provides point p of A' (preserving the language of the provides point)

- + extends to finite sets point wise and is denoted as:

$$A' \underset{\{m_j\}_{j \text{ in } J}}{+} \{B_j\}_{j \text{ in } J}$$

Formally speaking

Execution of activities

Paths

- An infinite sequence of states and transitions in the transition system of A in a repository Rep $[(A_0, q_0), \longrightarrow_0, (A_1, q_1), \longrightarrow_1, \ldots, (A_i, q_i), \longrightarrow_i, \ldots]$ is said to be a **path** iff

  for all $0 <= i < j$, $\longrightarrow_i = \xrightarrow[\text{Rep'}]{\text{I'}}$, $\longrightarrow_j = \xrightarrow[\text{Rep''}]{\text{I''}}$ with Rep' in Rep, Rep'' in Rep and $A_{i+1} = A_i \underset{\{m_j\}_{j \text{ in } J}}{+} \{B_j\}_{j \text{ in } J}$ such

  there is no $i<k<j$ such that $\longrightarrow_k = \xrightarrow[\text{Rep'''}]{\text{I'''}}$ with Rep''' in Rep then, Rep'' = Rep' / $\{B_j\}_{j \text{ in } J}$

Formally speaking

Execution of activities

Traces

- A path in the transition system of A in a repository Rep $[(A_0, q_0), \longrightarrow_0, (A_1, q_1), \longrightarrow_1, \ldots, (A_i, q_i), \longrightarrow_i, \ldots]$ is said to be a **trace** iff
  - there exists $0 <= i$, $\longrightarrow_i = \xrightarrow[\text{Rep'}]{\text{I'}}$ with Rep' in Rep such that there is no $k<i$ such that $\longrightarrow_k = \xrightarrow[\text{Rep''}]{\text{I''}}$ with Rep'' in Rep and Rep' = Rep, and
  - $A_0 = A$
- The set of all traces of a transition systems $\mathcal{S} = (S, \longrightarrow)$ is denoted as $O_{\mathcal{S}}$

Linear Temporal Logics and executions of activities

Linear Temporal Logics formulae
(defined as usual)

Let $\mathcal{V}$ be a set of proposition symbols, then the set of LTL formulae on $\mathcal{V}$, denoted as $LTLForm(\mathcal{V})$, is the smallest set S such that:

- $\mathcal{V} \subseteq S$, and
- if $\alpha, \beta \in S$, then $\{\neg\alpha, \alpha \vee \beta, \mathbf{X}\alpha, \alpha\mathbf{U}\beta\} \subseteq S$.

Linear Temporal Logics and executions of activities

Let α be an activity and *Rep* a repository, then if $\mathcal{S} = \langle S, \longrightarrow \rangle$ is the transition system for α and *Rep* then let $\pi = [(\alpha_0, q_0), \longrightarrow_0, (\alpha_1, q_1), \longrightarrow_1, \ldots]$ a path for $\mathcal{S}$. Let $\mathcal{V}$ be the set of actions in the signature of the *Rep* or in α, $\phi, \psi \in \mathit{LTLForm}(\mathcal{V})$, $a \in \mathcal{V}$ and $v \subseteq \mathcal{V}$ then:

- $\pi, v \models \mathbf{true}$,
- $\pi, v \models a$ iff $[a] \in [v]$,
- $\pi, v \models \neg\phi$ iff $\pi, v \not\models \phi$,
- $\pi, v \models \phi \vee \psi$ if $\pi, v \models \phi$ or $\pi, v \models \psi$,
- $\pi, v \models \mathbf{X}\phi$ iff $\pi_{[1]}, v_1 \models \phi$, and
- $\pi, v \models \phi \mathbf{U} \psi$ iff there exists $0 \leq i$ such that $\pi_{[i]}, v_i \models \psi$ and for all j, $0 \leq j < i$, $\pi_{[j]}, v_j \models \phi$

where $v_k = \bigcup_{\iota \in \iota_{k-1}} \iota$.

Linear Temporal Logics and executions of activities

Let α be an activity and *Rep* a repository, then if $\mathcal{S} = \langle S, \longrightarrow \rangle$ is the transition system for α and *Rep* then let $\pi = [(\alpha_0, q_0), \longrightarrow_0, (\alpha_1, q_1), \longrightarrow_1, \ldots]$ a path for $\mathcal{S}$. Let $\mathcal{V}$ be the set of actions in the signature of the *Rep* or in α, $\phi, \psi \in \mathit{LTLForm}(\mathcal{V})$, $a \in \mathcal{V}$ and $v \subseteq \mathcal{V}$ then:

Valuations are sets of labels, those that took the system from one state to another

- $\pi, v \models \mathbf{true}$,
- $\pi, v \models a$ iff $[a] \in [v]$,
- $\pi, v \models \neg\phi$ iff $\pi, v \not\models \phi$,
- $\pi, v \models \phi \vee \psi$ if $\pi, v \models \phi$ or $\pi, v \models \psi$,
- $\pi, v \models \mathbf{X}\phi$ iff $\pi_{[1]}, v_1 \models \phi$, and
- $\pi, v \models \phi\mathbf{U}\psi$ iff there exists $0 \leq i$ such that $\pi_{[i]}, v_i \models \psi$ and for all j, $0 \leq j < i$, $\pi_{[j]}, v_j \models \phi$

where $v_k = \bigcup\limits_{\iota \in \iota_{k-1}} \iota$.

Linear Temporal Logics and executions of activities

Let α be an activity and *Rep* a repository, then if $\mathcal{S} = \langle S, \longrightarrow \rangle$ is the transition system for α and *Rep* then let $\pi = [(\alpha_0, q_0), \longrightarrow_0, (\alpha_1, q_1), \longrightarrow_1, \ldots]$ a path for $\mathcal{S}$. Let $\mathcal{V}$ be the set of actions in the signature of the *Rep* or in α, $\phi, \psi \in LTLForm(\mathcal{V})$, $a \in \mathcal{V}$ and $v \subseteq \mathcal{V}$ then:

- $\pi, v \models \mathbf{true}$,
- $\pi, v \models a$ iff $[a] \in [v]$,
- $\pi, v \models \neg\phi$ iff $\pi, v \not\models \phi$,
- $\pi, v \models \phi \vee \psi$ if $\pi, v \models \phi$ or $\pi, v \models \psi$,
- $\pi, v \models \mathbf{X}\phi$ iff $\pi_{[1]}, v_1 \models \phi$, and
- $\pi, v \models \phi \mathbf{U} \psi$ iff there exists $0 \leq i$ such that $\pi_{[i]}, v_i \models \psi$ and for all j, $0 \leq j < i$, $\pi_{[j]}, v_j \models \phi$

where $v_k = \bigcup_{\iota \in \iota_{k-1}} \iota$.

Propositions are the labels of the transitions of the automata

Linear Temporal Logics and executions of activities

Let α be an activity and *Rep* a repository, then if $\mathcal{S} = \langle S, \longrightarrow \rangle$ is the transition system for α and *Rep* then let $\pi = [(\alpha_0, q_0), \longrightarrow_0, (\alpha_1, q_1), \longrightarrow_1, \ldots]$ a path for $\mathcal{S}$. Let $\mathcal{V}$ be the set of actions in the signature of the *Rep* or in α, $\phi, \psi \in \mathit{LTLForm}(\mathcal{V})$, $a \in \mathcal{V}$ and $v \subseteq \mathcal{V}$ then:

- $\pi, v \models \mathbf{true}$,
- $\pi, v \models a$ iff $[a] \in [v]$,
- $\pi, v \models \neg\phi$ iff $\pi, v \not\models \phi$,
- $\pi, v \models \phi \vee \psi$ if $\pi, v \models \phi$ or $\pi, v \models \psi$,
- $\pi, v \models \mathbf{X}\phi$ iff $\pi_{[1]}, v_1 \models \phi$, and
- $\pi, v \models \phi \mathbf{U} \psi$ iff there exists $0 \leq i$ such that $\pi_{[i]}, v_i \models \psi$ and for all j, $0 \leq j < i$, $\pi_{[j]}, v_j \models \phi$

where $v_k = \bigcup_{\iota \in \iota_{k-1}} \iota$.

Propositions are partitioned in clases of equivalence using the mappings of labels in the communication channels

Linear Temporal Logics and executions of activities

Let α be an activity and *Rep* a repository, then if $\mathcal{S} = \langle S, \longrightarrow \rangle$ is the transition system for α and *Rep* then let $\pi = [(\alpha_0, q_0), \longrightarrow_0, (\alpha_1, q_1), \longrightarrow_1, \ldots]$ a path for $\mathcal{S}$. Let $\mathcal{V}$ be the set of actions in the signature of the *Rep* or in α, $\phi, \psi \in \mathit{LTLForm}(\mathcal{V})$, $a \in \mathcal{V}$ and $v \subseteq \mathcal{V}$ then:

- $\pi, v \models \mathbf{true}$,
- $\pi, v \models a$ iff $[a] \in [v]$,
- $\pi, v \models \neg\phi$ iff $\pi, v \not\models \phi$,
- $\pi, v \models \phi \vee \psi$ if $\pi, v \models \phi$ or $\pi, v \models \psi$,
- $\pi, v \models \mathbf{X}\phi$ iff $\pi_{[1]}, v_1 \models \phi$, and
- $\pi, v \models \phi \mathbf{U} \psi$ iff there exists $0 \leq i$ such that $\pi_{[i]}, v_i \models \psi$ and for all j, $0 \leq j < i$, $\pi_{[j]}, v_j \models \phi$

where $v_k = \bigcup_{\iota \in \iota_{k-1}} \iota$.

As usual, the subindex denotes the
the suffix operator on paths

Linear Temporal Logics and executions of activities

Let α be an activity and *Rep* a repository, then if $\mathcal{S} = \langle S, \longrightarrow \rangle$ is the transition system for α and *Rep* then let $\pi = [(\alpha_0, q_0), \longrightarrow_0, (\alpha_1, q_1), \longrightarrow_1, \ldots]$ a path for $\mathcal{S}$. Let $\mathcal{V}$ be the set of actions in the signature of the *Rep* or in α, $\phi, \psi \in \mathit{LTLForm}(\mathcal{V})$, $a \in \mathcal{V}$ and $v \subseteq \mathcal{V}$ then:

- $\pi, v \models \mathbf{true}$,
- $\pi, v \models a$ iff $[a] \in [v]$,
- $\pi, v \models \neg\phi$ iff $\pi, v \not\models \phi$,
- $\pi, v \models \phi \vee \psi$ if $\pi, v \models \phi$ or $\pi, v \models \psi$,
- $\pi, v \models \mathbf{X}\phi$ iff $\pi_{[1]}, v_1 \models \phi$, and
- $\pi, v \models \phi \mathbf{U} \psi$ iff there exists $0 \leq i$ such that $\pi_{[i]}, v_i \models \psi$ and for all j, $0 \leq j < i$, $\pi_{[j]}, v_j \models \phi$

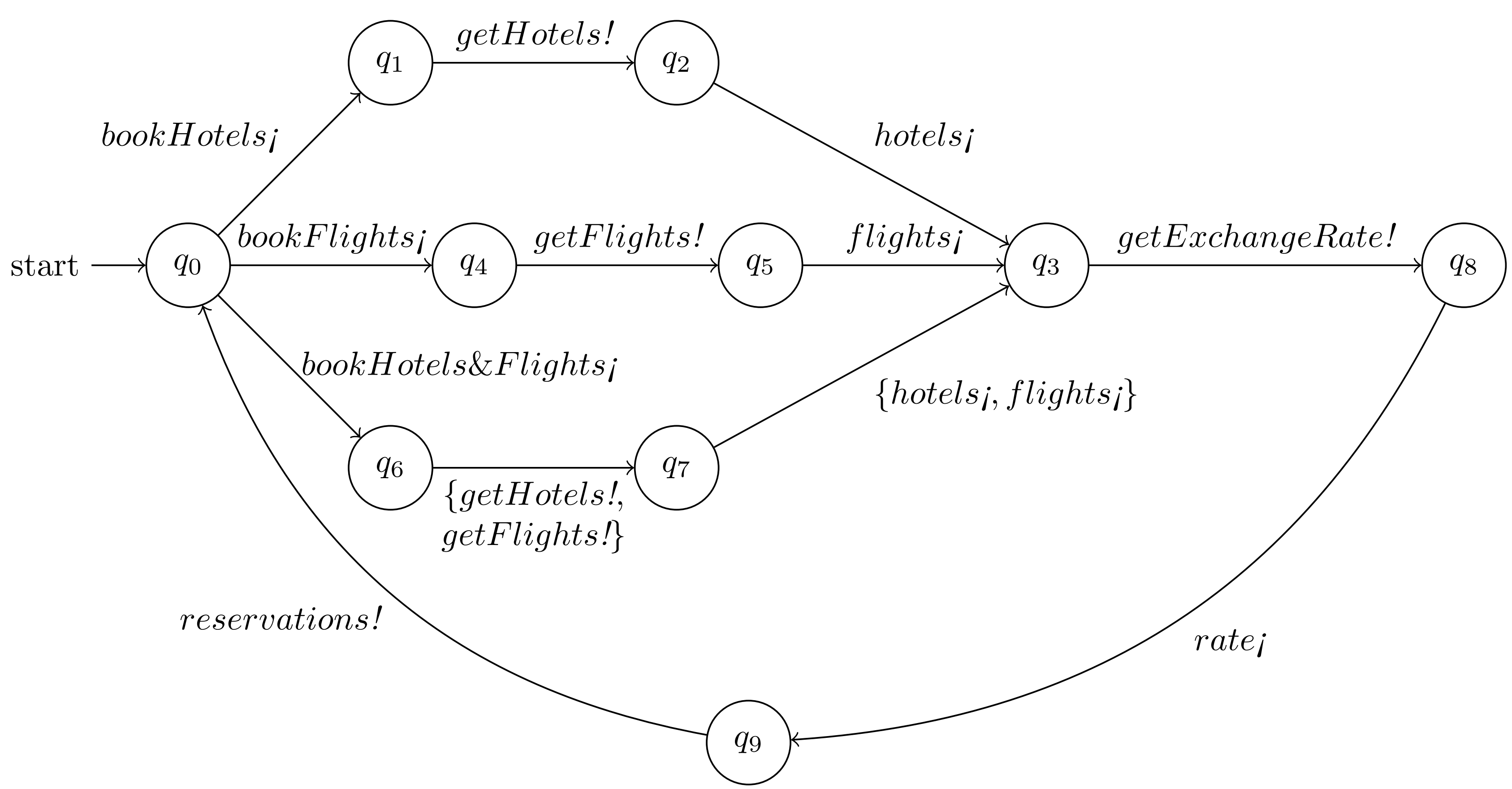
where $v_k = \bigcup_{\iota \in \iota_{k-1}} \iota$.

Linear Temporal Logics and executions of activities

Some examples of properties

Every execution of **TravelClient** requires the execution of **CurrenciesAgent**:

$$\text{For all } \pi \in O_{\mathcal{S}},\ \pi \models \Diamond \left(\bigvee_{a \in A_{M_{\mathsf{CurrenciesAgent}}}} a \right)$$

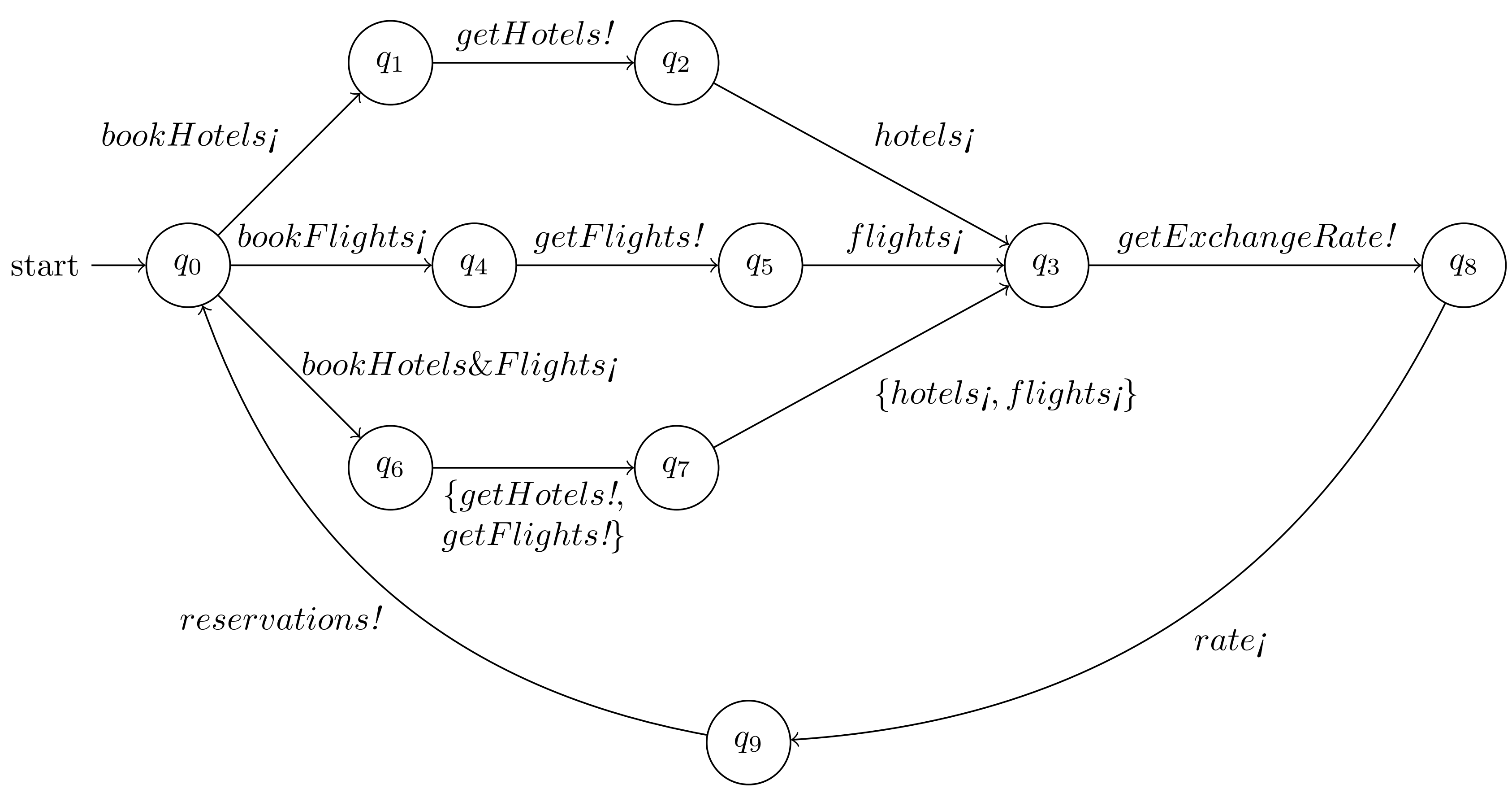


Linear Temporal Logics and executions of activities

Some examples of properties

There exists an execution of **TravelClient** that does not requires the execution of **FlightsAgent**:

$$\text{There exists } \pi \in O_{\mathcal{S}},\ \pi \models \Diamond \left(\neg \bigvee_{a \in A_{M_{\mathsf{FlightsAgent}}}} a\right)$$

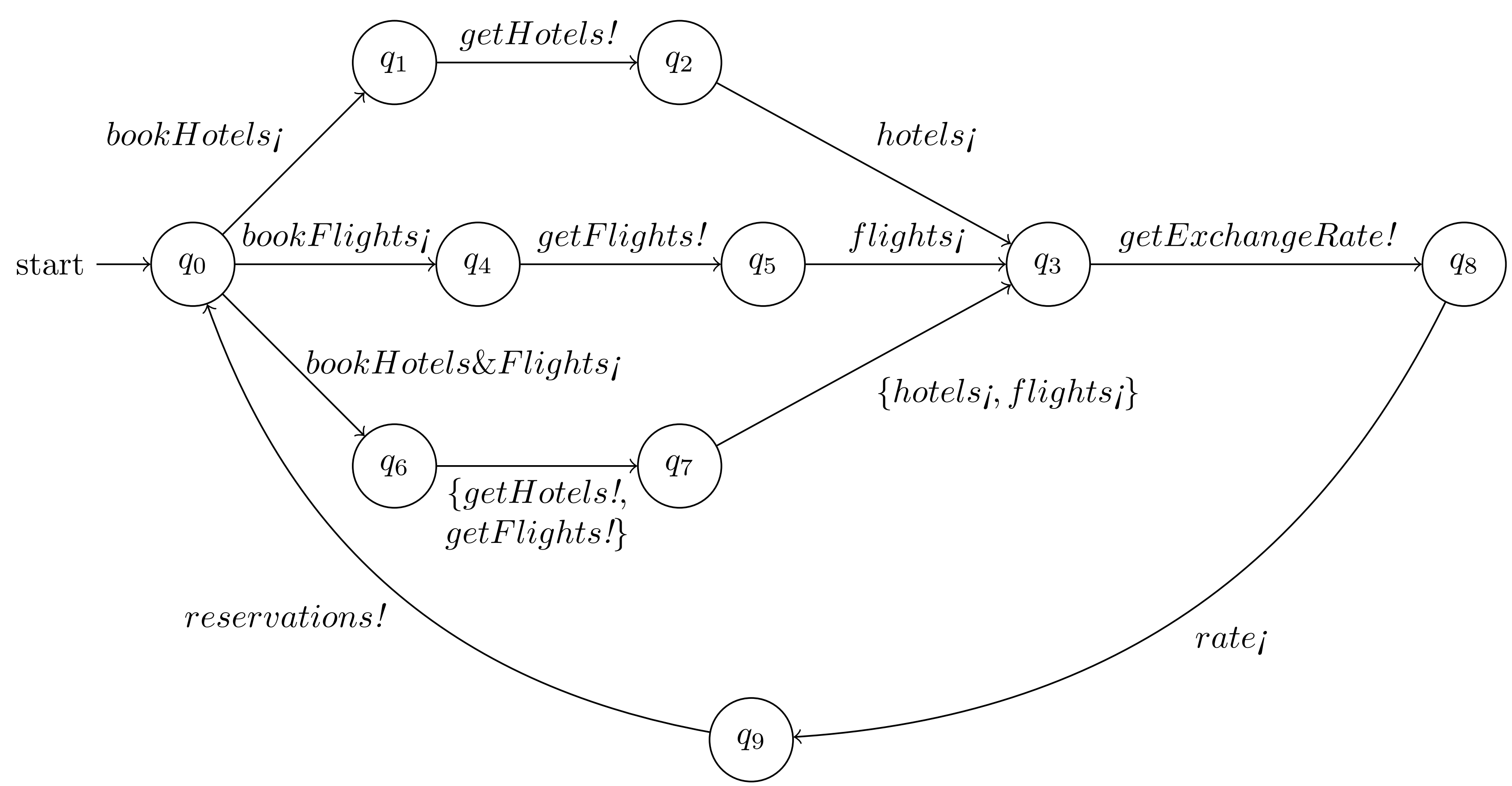


Linear Temporal Logics and executions of activities

Some examples of properties

Every execution of **TravelClient**, in the future, will receive an exchange rate and in the next state a reservation will issued:

$$\text{For all } \pi \in O_{\mathcal{S}},\ \pi \models \Box\,(hotels! \implies \Diamond\,(rate! \wedge \mathbf{X} reservation!))$$

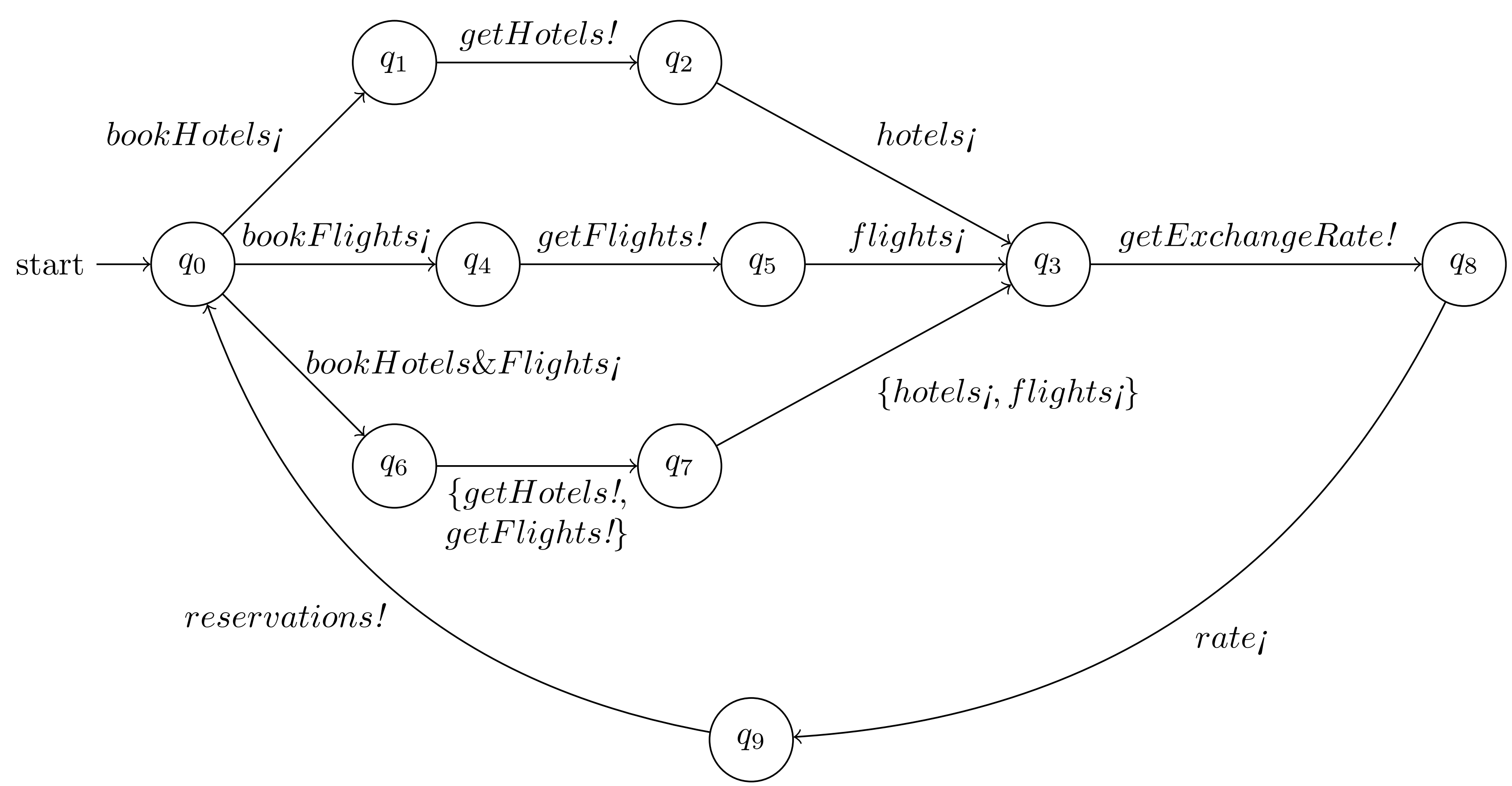


Linear Temporal Logics and executions of activities

Some examples of properties

In every execution of **TravelClient**, if we place an order for accommodation and flight, we will receive two reservations:

$$\text{For all } \pi \in O_{\mathcal{S}},\ \pi \models \Box\,(hotels\&Flights! \implies \Diamond\,(reservation! \wedge ((\neg hotels! \wedge \neg hotels\&Flights!)\,\mathbf{X} reservation!)))$$



Outro (Conclusions)

- We introduced an execution model for ARNs by providing an operational semantics based on a transition system

- We defined a linear temporal satisfaction relation between traces of the transition system and LTL formulae

Outro (Further work)

* Definition/implementation of a model-checking technique for analysing properties of ARNs using this semantics **[Fiadeiro, Ţuţu, Vissani, me]**

* Explore the incidence of different kinds of contracts as the means for formalising the negotiation of the Service Level Agreement (SLA)

* Implementation of a middleware capable of providing support for formal establishing of SLA as a part of the process of binding

The Encore

? & !

—8<———8<———8<———8<———8<———8<———8<—

Some other projects

- Use of global types and local graphs as a tool for negotiating the protocol to be used on the interfaces of ARN **[Tuosto, Vissani, me]**

- Analysis of a trace-based semantics for choreographies and global graphs **[Melgratti, Barbeito, me]**

- The formulation of a canonical proof-theoretic approach to model theory **[Maibaum, Chocrón, me]** and its extension to substructural logics **[Kurz, Maibaum, Chocrón, me]**