UNIVERSITÄT KOBLENZ · LANDAU

IWVI

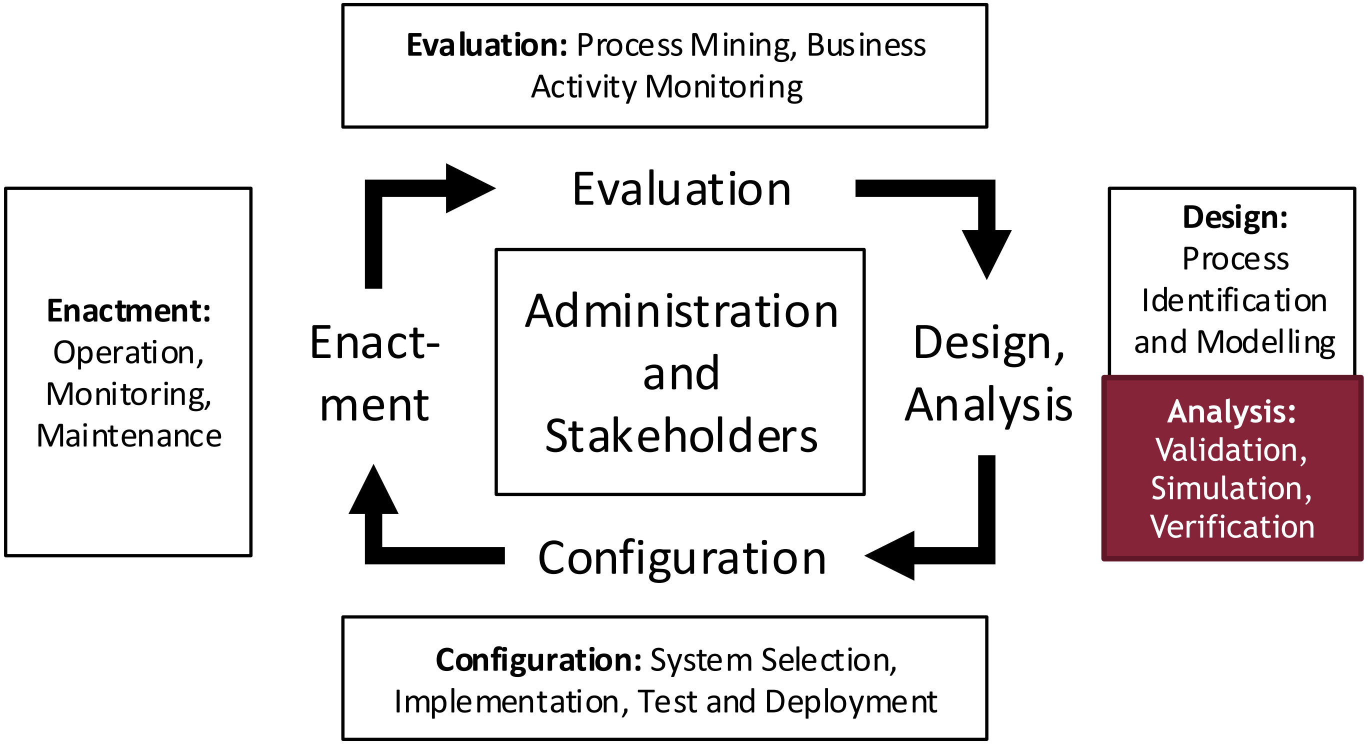
BUSINESS PROCESS MANAGEMENT

MODEL QUERY I: COMPUTATIONAL TREE LOGIC (CTL)

AGENDA

- Why Model Query?
- Model Checking with Temporal Logic
- Computational Tree Logic (CTL)
- Application of CTL

(PROCESS) MODELS IN THE BPM LIFECYCLE



Lecture

Model query I: CTL

- Analyze process models to find weaknesses and improvement potential (inefficiencies, law violations, etc.)

THE NEED FOR AUTOMATIC MODEL QUERY

7000 models

600 models

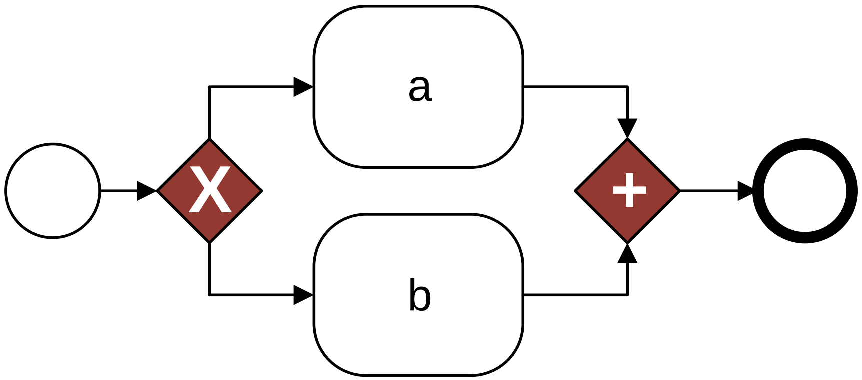
100 models

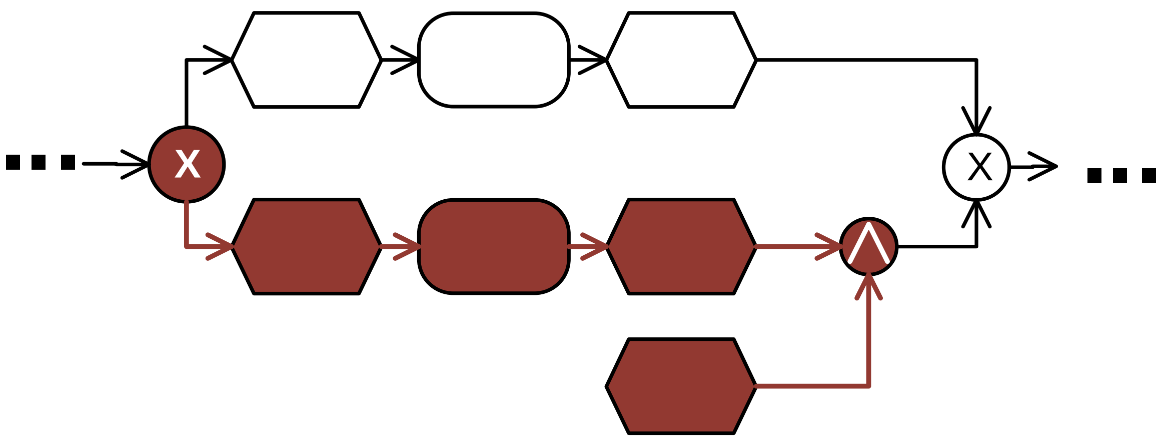
700 models

2200 models

EXEMPLARY PATTERNS FOR MODEL QUERY

CONFLICT DETECTION

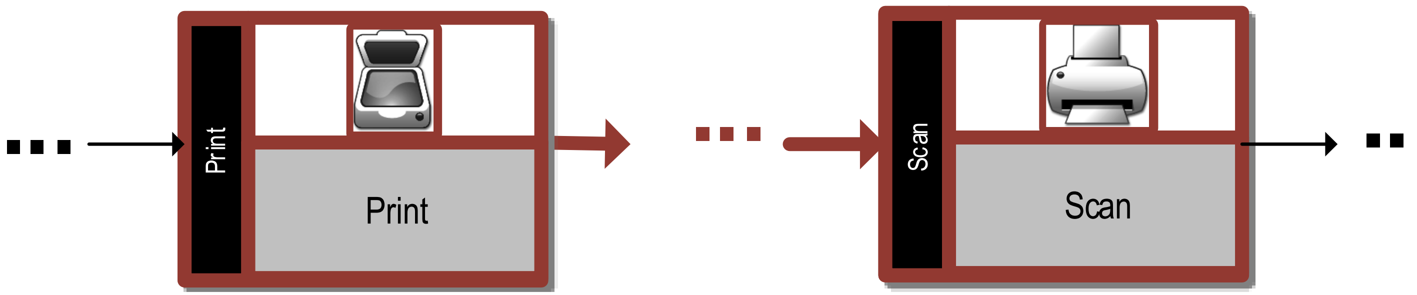


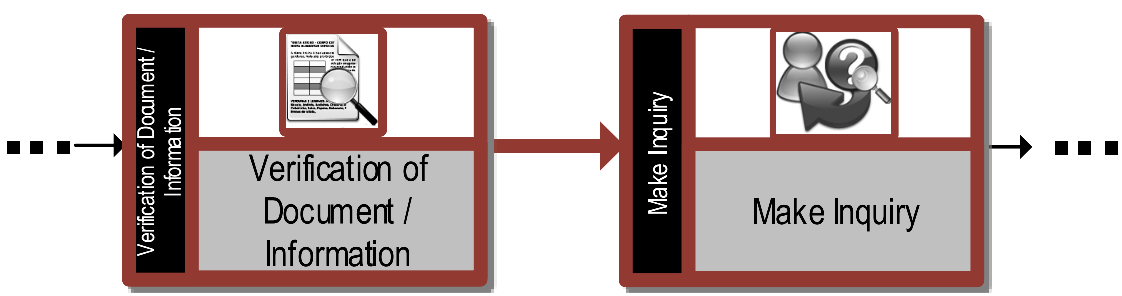


[Mendling 2007]

EXEMPLARY PATTERNS FOR MODEL QUERY

WEAKNESS DETECTION

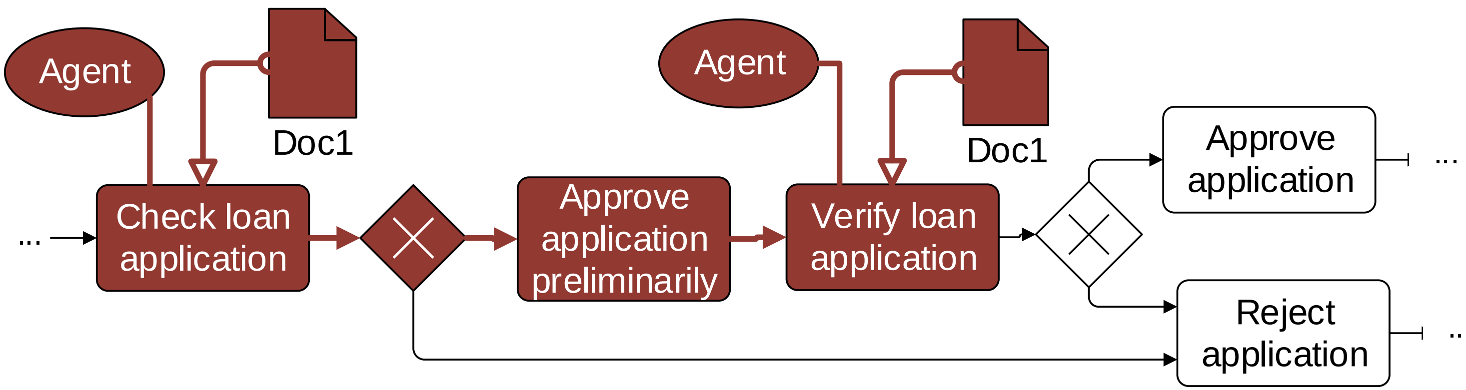




[Becker et al. 2010; 2012]

EXEMPLARY PATTERNS FOR MODEL QUERY

COMPLIANCE CHECKING



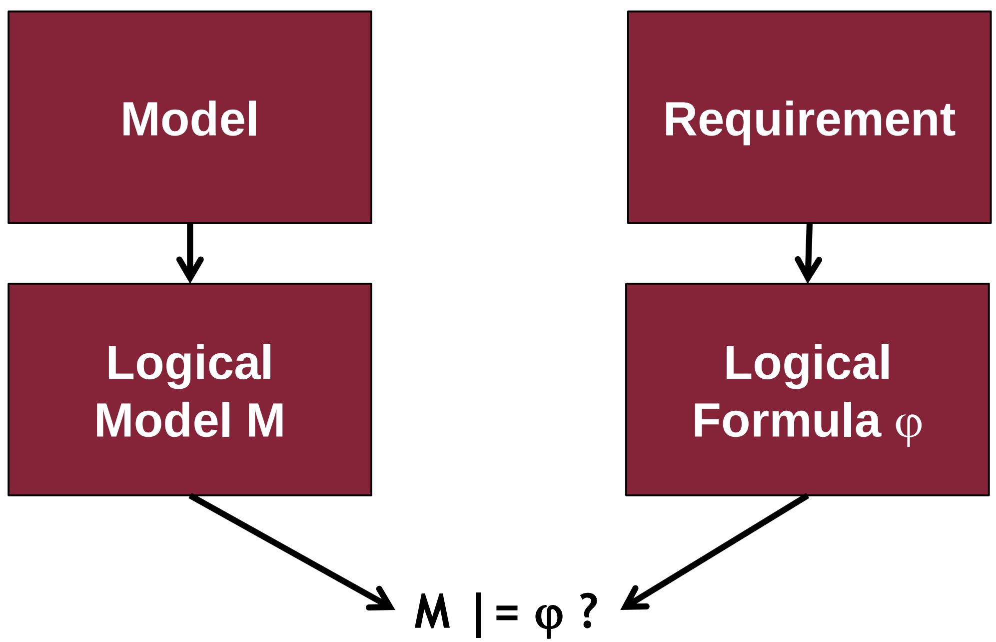
[Knorr and Stormer 2001; Awad et al. 2008]

AGENDA

- Why Model Query?
- Model Checking with Temporal Logic
- Computational Tree Logic (CTL)
- Application of CTL

MODEL CHECKING

- Given a (process) model M and a formula φ, model checking is the problem of verifying whether or not φ is true in M (written M |= φ)



[http://www.cs.ucl.ac.uk/staff/f.raimondi/]

TRANSITION SYSTEMS

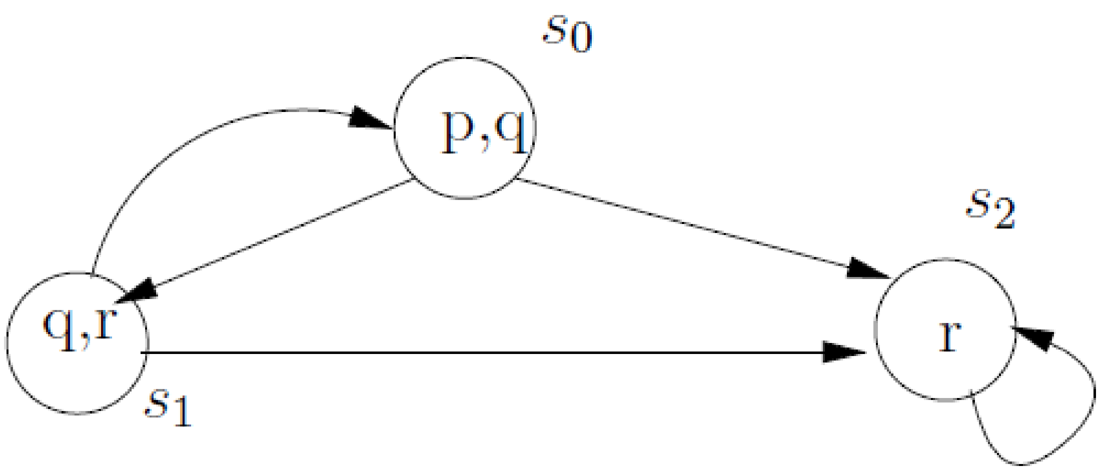
- A transition system represents a system (e.g., a process model) by means of states and transitions between states
- Transition system: $M=(S,R_t,L)$
- Set of states S
- Binary relation $R_t \subseteq S \times S$
- AP is a set of atomic propositions
- Labelling function $L: S \rightarrow \mathbb{P}(AP)$
- The relation R_t is serial, i.e., for every state $s \in S$, there exists a state s' such that sR_ts'

[http://www.cs.ucl.ac.uk/staff/f.raimondi/]

TRANSITION SYSTEMS

EXAMPLE

- $M=(S,R_t,L)$



- $S=\{s_0,s_1,s_2\}$
- $R_t=\{(s_0,s_1),\ (s_0,s_2),\ (s_1,s_0),\ (s_1,s_2),\ (s_2,s_2)\},$
- $L(s_0)=\{p,q\},\ L(s_1)=\{q,r\},\ L(s_2)=\{r\}$

[http://www.cs.ucl.ac.uk/staff/f.raimondi/]

TEMPORAL LOGIC

- Many model checking approaches are based mainly on temporal logic
- Various kinds of temporal logic:
  - Linear Temporal Logic (LTL),
  - Computational Tree Logic (CTL)
  - CTL*, …
- Here: CTL

[http://www.cs.ucl.ac.uk/staff/f.raimondi/]

AGENDA

- Why Model Query?
- Model Checking with Temporal Logic
- Computational Tree Logic (CTL)
- Application of CTL

CTL SYNTAX

- Set of atomic propositions $P=\{p_1, p_2, . . .\}$
- Atomic propositions stand for atomic facts which may hold in a model, e.g.
  - “check invoice”
  - “grant credit”
  - “employee”

- CTL Syntax:

$$\varphi ::= p_i \mid \varphi \mid \neg\varphi \mid \varphi\wedge\varphi \mid \varphi\vee\varphi \mid \varphi\rightarrow\varphi \mid AX\varphi \mid EX\varphi \mid AF\varphi \mid EF\varphi \mid AG\varphi \mid EG\varphi \mid A[\varphi U\varphi] \mid E[\varphi U\varphi]$$

[http://www.cs.ucl.ac.uk/staff/f.raimondi/]

CTL SEMANTICS

- Unary Operators
  - A: for All paths
  - E: there Exists a path
  - X: neXt state
  - F: in a Future state
  - G: Globally in the future
- Binary Operator
  - U: Until
- Example
  - $AG(p \rightarrow (EFq))$: “It is globally the case that, if p is true, then there exists a path from where we have found p such that at some point in the future q is true”

[http://www.cs.ucl.ac.uk/staff/f.raimondi/]

CTL SEMANTICS

- Let $M=(S,R_t,L)$ be a transition system.
- Let φ be a CTL formula and $s \in S$
- $M, s \models \varphi$ is defined inductively on the structure of φ:

- $M, s \models p$ iff $p \in L(s)$
- $M, s \models \neg\varphi$ iff $M, s \not\models \varphi$
- $M, s \models \varphi \wedge \psi$ iff $M, s \models \varphi$ and $M, s \models \psi$
- $M, s \models \varphi \vee \psi$ iff $M, s \models \varphi$ or $M, s \models \psi$
- $M, s \models AX\varphi$ iff $\forall s'$ such that sR_ts', $M, s' \models \varphi$
- $M, s \models EX\varphi$ iff $\exists s'$ such that sR_ts' and $M, s' \models \varphi$

[http://www.cs.ucl.ac.uk/staff/f.raimondi/]

CTL SEMANTICS

- $M, s_1 \models AG\varphi$ iff for all paths $(s_1, s_2, s_3, s_4, \ldots)$ such that $s_i R_t s_{i+1}$ and for all i, it is the case that $M, s_i \models \varphi$
- $M, s_1 \models EG\varphi$ iff there is a path $(s_1, s_2, s_3, s_4, \ldots)$ such that $s_i R_t s_{i+1}$ and for all i it is the case that $M, s_i \models \varphi$
- $M, s_1 \models AF\varphi$ iff for all paths $(s_1, s_2, s_3, s_4, \ldots)$ such that $s_i R_t s_{i+1}$, there is a state s_i such that $M, s_i \models \varphi$
- $M, s_1 \models EF\varphi$ iff there is a path $(s_1, s_2, s_3, s_4, \ldots)$ such that $s_i R_t s_{i+1}$, and there is a state s_i such that $M, s_i \models \varphi$

[http://www.cs.ucl.ac.uk/staff/f.raimondi/]

CTL SEMANTICS

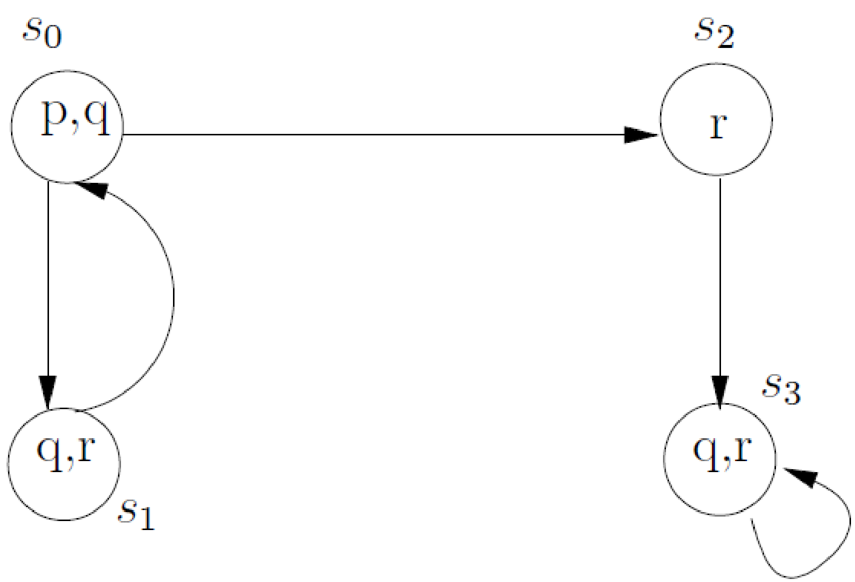
- $M, s_1 \models A[\varphi U \psi]$ iff for all paths $(s_1, s_2, s_3, s_4, \ldots)$ such that $s_i R_t s_{i+1}$ and there is a state s_j such that $M, s_j \models \psi$ and $M, s_i \models \varphi$ for all $i < j$.

- $M, s_1 \models E[\varphi U \psi]$ iff there exists a path $(s_1, s_2, s_3, s_4, \ldots)$ such that $s_i R_t s_{i+1}$ and there is a state s_j such that $M, s_j \models \psi$ and $M, s_i \models \varphi$ for all $i < j$.

- We write $M \models \varphi$ if a formula is true in all states of a model.

[http://www.cs.ucl.ac.uk/staff/f.raimondi/]

EXAMPLES



- M, s_0 |= EX(¬p)
- M, s_0 |= EX(r)
- M, s_1 |= AG(q ∨ r)
- M, s_2 |= A[rUq]
- M, s_0 |= A[pUq]
- M, s_1 |= E[qUEG(r)]
- M, s_0 |= ¬AG(q)
- M, s_1 |= EF(AG(q))

[http://www.cs.ucl.ac.uk/staff/f.raimondi/]

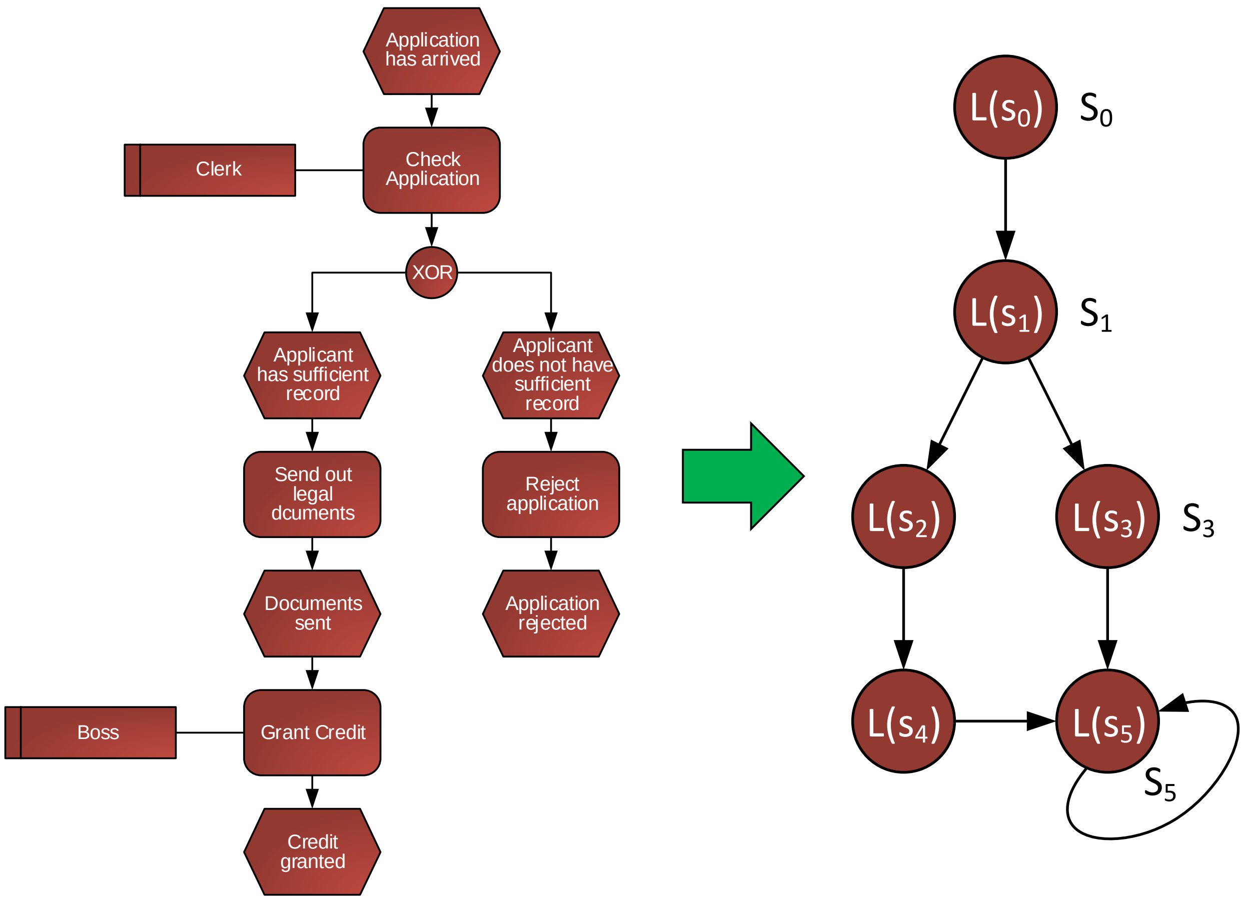
AGENDA

- Why Model Query?
- Model Checking with Temporal Logic
- Computational Tree Logic (CTL)
- Application of CTL for BPM

CONCEPTUAL PROCESS MODELS AND CTL

- How can we use CTL to analyze (process) models?
- Step 1: transform a model into a transition system
  - activities → states
  - Events → ∅
  - labels → atomic propositions
  - annotated objects → atomic propositions
- Step 2: create CTL formulas representing, e.g., compliance rules or weakness patterns
- Step 3: check the transition system with a CTL model checker

COMPLIANCE EXAMPLE



- L(s0)={start}
- L(s1)={n_check_application, o_clerk}
- L(s2)={n_send_out_legal_documents}
- L(s3)={n_reject_application}
- L(s4)={n_grant_credit, o_boss}
- L(s5)={end}

COMPLIANCE EXAMPLE

- Rule 1: Legal Documents have to be sent to the customer, before a credit can be granted

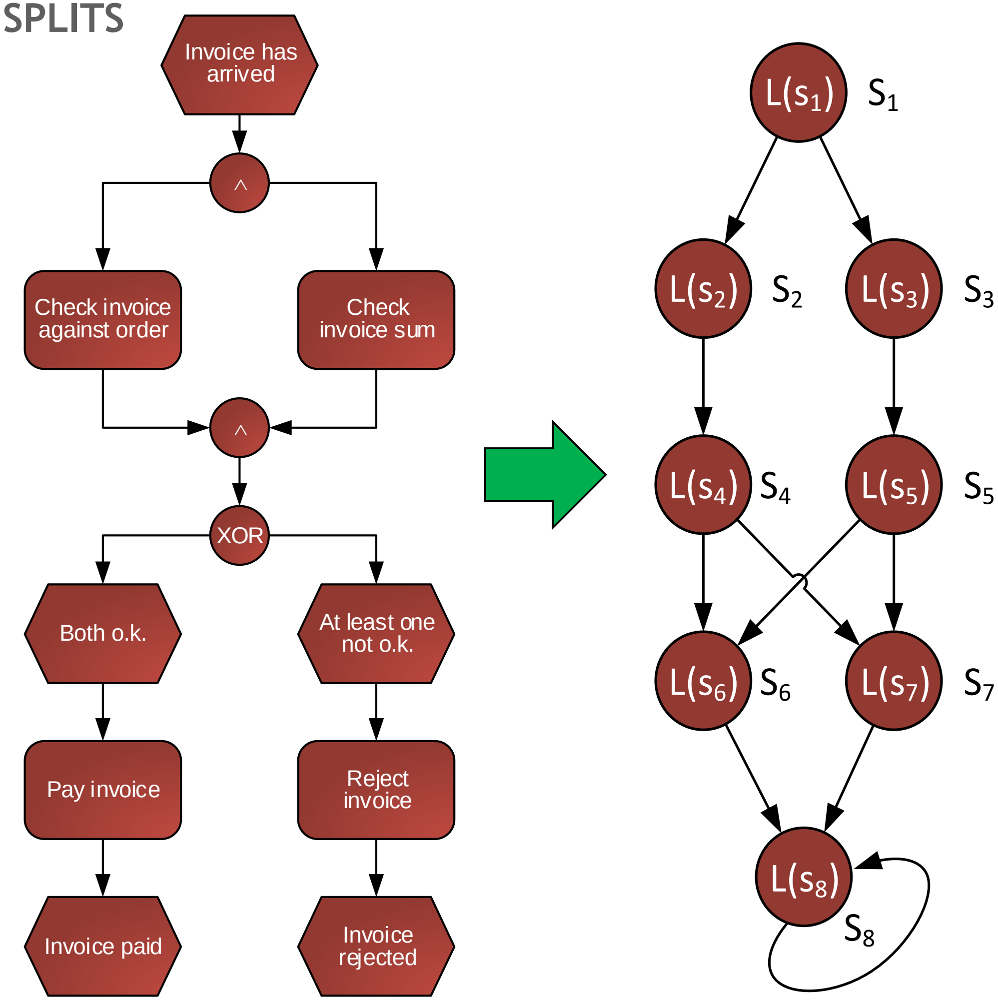
- $M, s_0 \models A[(\neg n\_grant\_credit) U n\_send\_out\_legal\_documents]$

- Rule 2: Credits can only be granted by the boss

- $M, s_0 \models \neg EF(n\_grant\_credit \wedge \neg o\_boss)$

CAUTION!

AND SPLITS



- L(s1)={start}
- L(s2)={Check_invoice_against_order}
- L(s3)={Check_invoice_sum}
- L(s4)={Check_invoice_sum}
- L(s5)={Check_invoice_against_order}
- L(s6)={Pay_invoice}
- L(s7)={Reject_invoice}
- L(s8)={end}

KEY LEARNINGS

- BPMN models can be transformed into a transition system (M)
- (Legal) regulations can be transformed into formulas
- For compliance checking, queries are ALWAYS formulated starting in s_0

UNIVERSITÄT KOBLENZ · LANDAU

BUSINESS PROCESS MANAGEMENT

MODEL QUERY 1: COMPUTATIONAL TREE LOGIC (CTL)

INSTITUTE FOR IS RESEARCH