UNIVERSITÄT KOBLENZ · LANDAU

BUSINESS PROCESS MANAGEMENT - EXERCISE

COMPUTATIONAL TREE LOGIC (CTL)

CTL SYNTAX AND SEMANTICS

- Set of atomic propositions $P=\{p_1, p_2, . . .\}$
- CTL Syntax:

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

- 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

(for a detailed definition see lecture slides)

CTL SEMANTICS: IMPLICATION

- 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

- $M, s \models \varphi \rightarrow \psi$ iff $M, s \not\models \varphi$ or $M, s \models \psi$

- Example:
  - $x \rightarrow y$ (if x is true, then y is true too)
  - $\neg x \vee y$ (either x is false, or y has to be true)

CTL OPERATORS

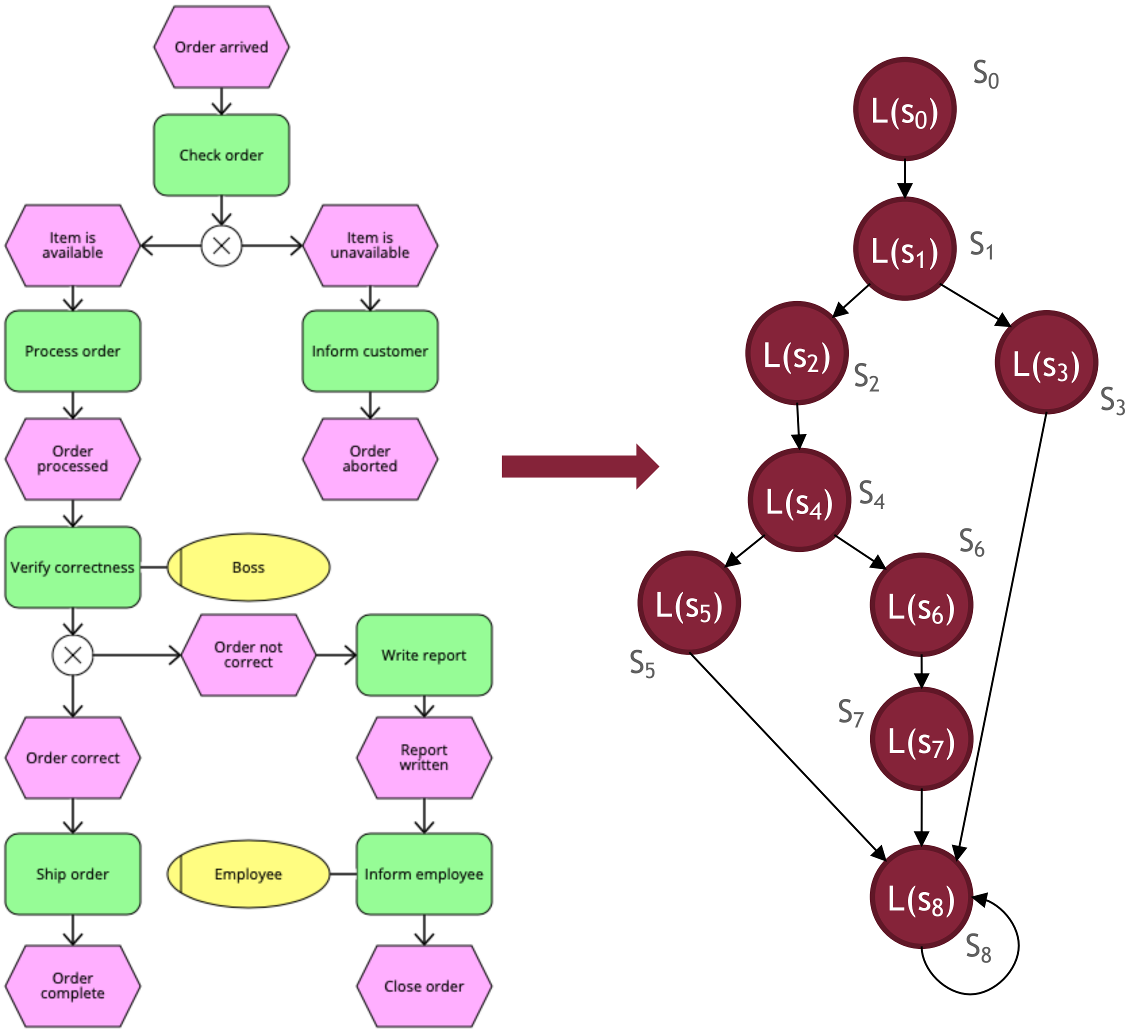
A (for all paths)	X (in the next state)	AXφ
	F (in a future state)	AFφ
	G (globally in the future)	AGφ
	U (until)	A[φUφ]
E (there exists a path)	X	EXφ
	F	EFφ
	G	EGφ
	U	E[φUφ]

- If we don't use operators, then we are always talking about the current state

CTL OPERATORS - EXAMPLES

- $M, S_0 \models x \rightarrow y$
  - (if x is true in S_0, then y is also true in that same state)
- $M, S_0 \models x \rightarrow EFy$
  - (if x is true in S_0, then somewhere in the future, y is also true)
- $M, S_0 \models EF(x \rightarrow y)$
  - (if somewhere in the future x is true, then y is also true in that same state
- $M, S_0 \models EF(x \rightarrow EFy)$
  - (if somewhere in the future x is true, then somewhere in the future (from x), y is also true)

EPC TO TRANSITION SYSTEM



- $L(s_0)$={start}
- $L(s_1)$={n_check_order}
- $L(s_2)$={n_process_order}
- $L(s_3)$={n_inform_customer}
- $L(s_4)$={n_verify_correctness, o_boss}
- $L(s_5)$={n_ship_order}
- $L(s_6)$={n_write_report}
- $L(s_7)$={n_inform_employee, o_employee}
- $L(s_8)$={end}

COMPLIANCE CHECKING USING BUSINESS RULES

- Step 1: transform a model into a transition system
- Step 2: create CTL formulas representing, e.g., compliance rules or weakness patterns
- Step 3: check the transition system with a CTL model checker

- Important: You have to assume, that you don't know where potential problems can be found in the model, so in this case model checking always has to start in S_0

BUSINESS RULES

- The order has to be verified, before it can be shipped

  x before y

- The order has to be verified by the boss

  x and y

- If the order is not correct, an employee has to be informed about this

  if x then y

- If the customer is informed that the product is unavailable, the order needn’t be verified by the boss (to save time)

  if x then y

BUSINESS RULES

- The order has to be verified, before it can be shipped (x before y)

→ y must not occur, until x has occurred

- $M, S_0 \models A[\neg y \; U \; x]$

- $M, S_0 \models A[x \; U \; y]$

BUSINESS RULES

- The order has to be verified by the boss (x and y)

→ if we have x, y also has to occur in that same state

- $M, S_0 \models \neg EF(x \wedge \neg y)$
- $M, S_0 \models AG(x \rightarrow y)$ or $M, S_0 \models AG(\neg x \vee y)$

- $M, S_0 \models \neg EF(x \rightarrow \neg y)$ or $M, S_0 \models \neg EF(\neg x \vee \neg y)$

- $M, S_0 \models AG(x \wedge y)$

BUSINESS RULES

- If the order is not correct, an employee has to be informed about this (if x then y)

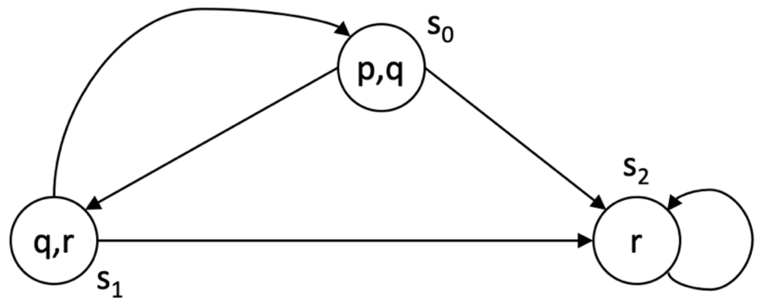
→ if x occurs, then somewhere in the future y has to occur

- $M, S_0 \models \neg EF(x \wedge \neg EFy)$
- $M, S_0 \models AG(x \rightarrow EFy)$ or $M, S_0 \models AG(\neg x \vee EFy)$

- $M, S_0 \models \neg EF(x \rightarrow \neg EFy)$ or $M, S_0 \models \neg EF(\neg x \vee \neg EFy)$

- $M, S_0 \models \neg EF(x \wedge \neg y)$
- $M, S_0 \models AG(x \wedge EFy)$
- $M, S_0 \models EF\ (x\ U\ y)$

COMPLIANCE CHECKING



- M, s_1 |= EX(¬p) → true
- M, s_1 |= AX(¬p) → false
- M, s_1 |= AG(q V r) → true
- M, s_0 |= A [q U r] → true

TRANSITION SYSTEM: LABELING FUNCTION

- Transition system: $M=(S,R_t,L)$
- Set of states S
- Binary relation $R_t \subseteq S \times S$
- Labeling function $L: S \longrightarrow 2^{AP}$
  - Input: Elements of S (s_1, s_2)
  - AP (atomic propositions) = {A,B,C}
  - L maps each state S to a set of atomic propositions
- $2^{AP} \rightarrow 2^3 = 8$
- Possible labels: {A}, {B}, {C}, {AB}, {AC}, {BC}, {ABC}, {}

UNIVERSITÄT KOBLENZ · LANDAU

IWVI

BUSINESS PROCESS MANAGEMENT - EXERCISE

INSTITUTE FOR IS RESEARCH

www.uni-koblenz.de