ACSR과 STATECHART를 이용한 소프트웨어 명세

고려대학교
정형기법 연구실
황대연
dyhwang@formal.korea.ac.kr

2011. 1. 7

정형명세

- Statechart
- Petri Net
- Scade
- CCS
- CSP
- ACSR
- Z notation
- B method
- Coq
- SMV
- SPIN

.

.

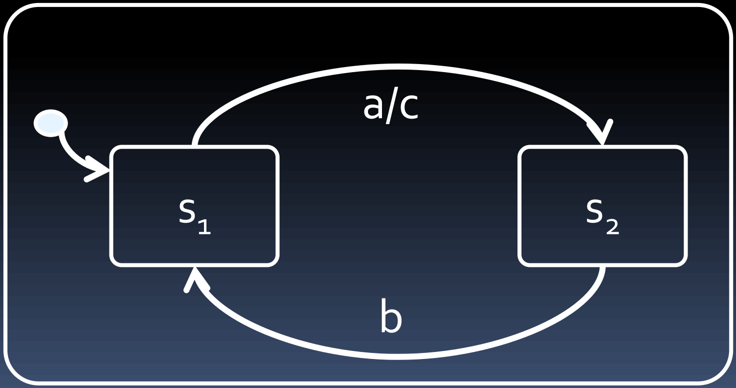
.

정형명세

Statechar
Petri Net
Scade
CCS
CSP
ACSR
Z notation
B method
Coq
SMV
SPIN
.
.
.

Statechart

- 시각적 정형화
- 상태에 기반한 행위 명세
- UML 포함



ACSR

- 프로세스 대수 기반
- 자원 + 우선 순위
- 시간

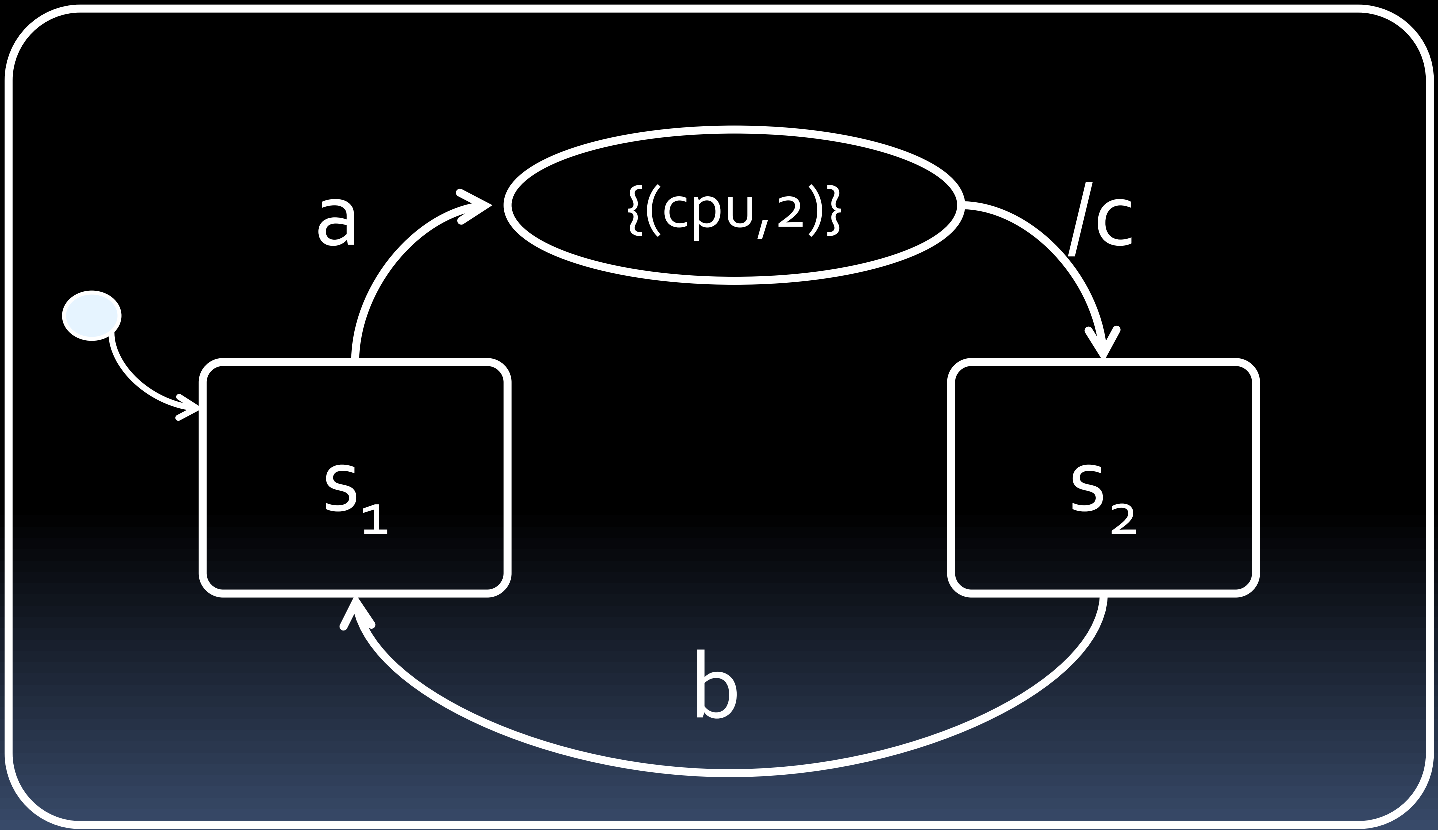
$$P = (a,1).Q + \{(cpu,1)\}:P'$$

Statechart + ACSR

TRoS

Timed and Resource Oriented Statecharts

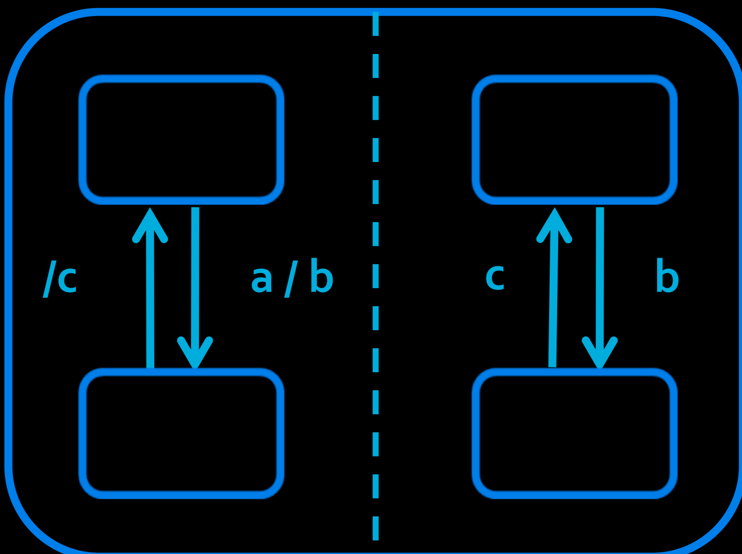
TRoS



Statechart || ACSR

- 시간
- 자원
- 통신
- 우선 순위

예 제



Control = P|Q|R

P = (a_s!, 1).P′ + { }:P
P′ = (e! , 1).{(cpu,1)}:P + { }:P′

Q = (e?,1).Q′ + { }:Q
Q′ = (c_s?, 1).Q + { }:Q′

R = (c_s?, 2).R + { }:R

	Statechart	event	ACSR
0	S1, S3		P,Q,R
1	S1, S3	a	P′, Q, R
2	S2, S3	b	P'_1, Q′, R
3	S1, S4	c	P'_1, Q′, R
4	S1, S3		P'_1, Q, R
5	S1, S3	{ }	P, Q, R

Step Algorithm

Definition 3.1 (Preemption Relation) *For two actions, α, β, we say that β preempts α ($\alpha \prec \beta$), if one of the following cases hold:*

(1) Both α and β are events in $\mathcal{D}_E$, where $\alpha = (a, p)$, $\beta = (a, p')$, and $p < p'$

(2) Both α and β are actions in $\mathcal{D}_R$, where

$$\begin{aligned}
&(\rho(\beta) \subseteq \rho(\alpha)) \wedge \\
&(\forall (r,p) \in \alpha \,.\, (((r,p') \in \beta \Longrightarrow p \leq p') \wedge ((r,p') \notin \beta \Longrightarrow p = 0))) \wedge \\
&(\exists (r,p') \in \beta \; \exists (r,p) \in \alpha \,.\, p < p')
\end{aligned}$$

(3) $\alpha \in \mathcal{D}_R$ and $\beta \in \mathcal{D}_E$, with $\beta = (\tau, p)$ and $p > 0$. □

Step Algorithm

Definition 3.1 (Preemption Relation) *For two actions, α, β, we say that β preempts α ($\alpha \prec \beta$), if one of the following cases hold:*

(1) Both α and β are events in $\mathcal{D}_E$, where $\alpha = (a, p)$, $\beta = (a, p')$, and $p < p'$

(2) ... $\wedge \ ((r, p') \notin \beta \Longrightarrow p = 0))) \ \wedge$

(3) ... > 0. □

Algorithm 4.1: NextStep(C,I)

Input : A configuration C, a set of input events I
Output : A set of transitions T

$T := \emptyset$;
while ($T \subset addToStep(C, I, T)$) **do**
Choose a transition t with highest priority from ($addToStep(C, I, T)$ - T);
Add t to T;
Return T;

Step Alg

Algorithm 4.4: *StepExecTmAct(C)*

Input : A configuration C
Output : A configuration C, a set of input events I

N is the timed configuration of C;
$T = NextStepTmAct(N)$;
$I' = \emptyset$;
for every $t \in T$ **do**
    Add newly generated events into I' by executing actions in t;
    Delete the source node of t from C;
    Add the target node of t to C;
$I = I' \cup I$;
CLK = CLK + 1;
Add the occurred timed event to I;
Return C and I;

Algorithm 4.3: *NextStepTmAct(N)*

Input : A timed configuration N
Output : A set of enabled transitions T

$T = \emptyset$;
Compute a set TA of consistent sets of timed actions;
Compute a set TAP of sets of timed actions that are not preempted by any set in TA;
Choose a set H from TAP;
for every a in H **do**
    Add t to T;
Return T;

Al... ...les;

while $(I \neq \emptyset)$ **do**
    $(C, I) = StepExec(C, I)$;
$(C, I) = StepExecTmAct(N)$;

Add newly generated events into I' by executing actions in t;
Delete the source node of t from C;
Add the target node of t to C;
$I = I'$
Reset the relevant timers if exits;
Return C and I;

$T := \emptyset$;
while $(T \subset addToStep(C, I, T))$ **do**
    Choose a transition t with highest pri...
    from $(addToStep(C, I, T)$ - $T)$;
    Add t to T;
Return T;

(3)

$$addToStep(T) = En(C, I) \cap consTrans(T)$$

연구 진행

- 적용
- 검증 및 검증 도구

?

기존 연구 소개

- Using a Process Algebra to control B OPERATIONS, 1999 Integated Formal Methods - Helen Treharne
- How to combine Z with Process Algebra, 1998 – C. Fischer
- A combination of Object-Z and CSP, 1997 – C. Fischer
- Specification of an Access Control System with a Formalism Combining CCS and CASL – 2002 Gwen Salaun
- Casl – Chart : A Combination of Statecharts and of the Algebraic Specification language Casl – 2000 G. Reggio
- Combining csp and b for specification and property verification M. Butler, 2005 In Proceedings of Formal Methods
- TROS || ACSR, 2010 Jin Hyun Kim