NuSRS 2.0 사용자 매뉴얼 (ver.1.0)

http://dependable.kaist.ac.kr/~nusrs nusrs@dependable.kaist.ac.kr

Dependable Software lab. KAIST 한국과학기술원

Software Engineering Lab. Deptpartment of Computer Science





- ♦ NuSRS 2.0 소개
 ♦ 개발 및 실행 환경
- ◆ NuSRS 2.0 기능 구성
- ◆ 화면 구성
- ◆ NuSRS 2.0 기능 상세 설명
 - 메뉴별 기능, 툴바
 - Hierarchy Window
 - Description Window
 - ▶ FOD, FSM(TTS), SDT 편집창
 - Type Window
 - Model Checking using SMV



NuSRS 2.0 소개

◆ NuSCR로 기술된 요구사항의 명세 및 검증을 지원하는 자동화 도구

◆ NuSRS 2.0의 대표적 개선점

- 사용자 편의성 최대화
 - ▶ Edit 메뉴 (Copy, Paste, Delete) 추가
 - ➢ View 메뉴 (Zoom In, Zoom Out, Zoom Selected, Fit to Window, Auto Align) 추가
- 보다 많은 정보의 가시화
 - ▶ Description, Type Window, Console Window 추가
- 편집시 철저한 일관성 검사
- 사용자의 입력오류를 최대한 방지하기 위한 자동생성 기능 및 오류 메시지 기능 강화
- 사용자에게 친숙한 인터페이스 구현













메뉴별 기능, 툴바



KAIS1

Software Engineering Lab. Deptpartment of Computer Science



메뉴 > File

♦ File

- New-새로운 화일 생성
- Open 불러오기
- Close File 파일 닫기
- Save 저장하기
- Save As 다른 이름으로 저장하기
- Save Without Graphical Info
 그림 정보 없이 저장하기
- Save as Image
 그림으로 저장하기
- ▶ *Print* 프린트하기
- ▶ *Exit* 프로그램 종료



NuSRS Editor 2.0		
File Edit View Window Verifi	cation Help	
📄 New		
🗟 Close File		
📙 Save		
🛃 Save As		
Save Without Graphical Info		
Construction of the second sec	-	
	-	
N EXIC]	
Description Window		
		Type V



♦ Edit

- ▶ *Cut* 선택된 영역을 잘라내기
- Copy 선택된 영역을 복사
- Paste 복사된 영역을 붙여넣기
- Delete 선택된 영역을 삭제
- Select All 현재 창에 있는 모든 객체를 선택





메뉴 > View

View

- FIND 찾기
- View IO Vars I/O Variable을 보여줌
- ▶ *Zoom In* 확대
- ▶ Zoom Out 축소
- Zoom Selected 선택된
 영역을 확대
- Fit to Window 윈도우
 사이즈에 맞게 확대 또는
 축소
- ▶ Auto Align 자동 배치





메뉴 > Window

Window

- Close 현재 편집중인 창을 닫아줌
- Close All Windows 현재
 열려 있는 모든 창을 닫아줌







♦ Help ▶ About – NuSRS 프로그램의 정보를 보여줌









불러오기 (1/3)

KAIS

1. 메뉴에서 File > Open을 선택한다

NuSRS에서 작성된 .xml 확장자 를 가진 파일 대상

NuSRS Editor 2.0 File Edit View Window Verification	ت ^د کا Heln	
New Den		
Close File Save Save Save As Save Without Graphical Info Save as Image Print Save Exit		
Description Window		
	Type Window Console Window	

불러오기 (2/3)

대화상자가 열리면 불러오고 싶은 파일을 선택한다.

🔲 NuSRS Editor 2.0 - D:\LAB PR	OJECTNuSRS - 합친거test.xml	
File Edit View Window Ver	ification Help	
📲 Hierarchy Window		
	🗟 open XML File 🗵	
	검색위치: 💼 NuSRS - 합친기 🔹 🖬 🔂 🗂 🔡 🔚	
	d.xml	
	🗋 dd.xml	
	☐ test.xml	
	nal-Verification.xml test_1227-16-40.xml	
	nal-Verification 1219-18-04.xml	
Desctiprion Window		vml
		XIIII
		으 기저
		트 시 (S)
	Tumo Méndous 💭 Coursels Méndous	
		16
	۱۱ ^۳	



저장하기

1. 메뉴에서 File > Save 를 선택한다. 현재 작업중인 파일이 저장된다.





다른 이름으로 저장하기 (1/3)

1. 메뉴에서 File > Save As를 선택한다.





다른 이름으로 저장하기 (2/3)

 대화상자가 열리면 저장하고 싶은 파일명을 입력한 뒤 저장을 클릭한다. 현재 작업중 파일이 저장된다.

📄 NuSRS Editor 2.0 - D:\LAB PROJECT\NuSRS - 합친거\test.xml	r 🛛 🛛	í l
File Edit View Window Verification Help		
	*	
Hierarchy Window		
Root Save XML File		
Root	h_1 f_17 h_3 th_5 f_5	Test1.xml 로 저장
		20

그래픽 정보 없이 저장하기 (1/2)

1. 메뉴에서 File > Save Without Graphical Info를 선택한다.





그래픽 정보 없이 저장하기 (2/2)

 대화상자가 열리면 저장하고 싶은 파일명을 입력한 뒤 저장을 클릭한다. 현재 작업 중 파일이 그래픽 정보를 제외하고 저장된다.

	💼 NuSRS Editor 2.0 - D:LAB PROJECT'NuSRS - 합친거ttest.xml 🛛 🗖 🛛	
	File Edit View Window Verification Help	
	New Image: Second	
	Save Save As Save Without Graphical Save as Image Print 제장 위치 NuSRS-1 Litt Litt	
	Image: CVS Test3.xml Image: CVS Test3.xml <th>사용자 지정</th>	사용자 지정
KAIST	Image: TemplateNumber Image: Test3-NoGraphicsInfo.xml <	이름.xml도 저장

이미지로 저장하기 (1/2)

1. 메뉴에서 File > Save as Image를 선택한다.





이미지로 저장하기 (2/2)

 대화상자가 열리면 저장하고 싶은 파일명을 입력한 뒤 저장을 클릭한다. 현재 작업 중 파일이 이미지로 저장된다.

	🔲 NuSRS Editor 2.0 - D:\LAB PF	iOJECT/NuSRS - 합친거/test.xml 📫 🖸 🔀	
	File Edit View Window Ve	rification Help	
	2 🗳 🔒 🤹 🕓	* 🖹 🖾 🗔 冬 🔍 🔯 💷 💽 🚬 🎽	
	Real Hierarchy Window		
	P- 🚍 Root └- 🗋 g_abc	📓 Save current diagram to image file 📉 🔀	
		제장위치 💼 NuSRS - 합친거 🔍 🝙 🚖 💼 😁	
		data	
		h_1	
		h_3	사용자 지정
	E Description Window		이르 ᢦᡨ근
	P- ☐ Root		이금.지미도
			저장
	P C TemplateNumber	저장 취소	
	P □ Input		
	f_2 : boolean	Type Window 🖉 Console Window	
	📕 — 🗋 f_3 : boolean	f_1:boolean	
	f_4 : boolean	f_17 : boolean	
LAIST		f_2:boolean	
			24

프린트하기

1. 메뉴에서 File > Print를 선택한다.





편집 > 잘라내기 (1/2)

1. 마우스 왼쪽 클릭 또는 drag & drop으로 잘라내고자 하는 영역을 선택한다.

NuSRS Editor 2.0 - D:'LAB PROJECT'NuSRS	- 합친기itest.xml	• ø 🛛
File Edit View Window Verification Hel	p	
2 🗳 🛃 🚳 🔓 🗎		
Rierarchy Window	O Root ×	
P □ Root □ g_abc	$\begin{array}{c} \hline \\ \hline $	_1 173
	$\xrightarrow{L_4}$	×
f_2 : boolean f_3 : boolean f 4 : boolean	f_1: boolean	
Output	f_2:boolean	-



편집 > 잘라내기 (2/2)

Edit > Cut 을 선택한다. 선택된 영역이 잘라내어 진다. (Ctrl + X 로도 가능)







편집 > 복사 (2/2) Edit > Copy 를 선택한다. 선택된 영역이 2. 복사된다.(Ctrl + C 로도 가능) ۲ ۵ ک 🛅 NuSRS Editor 2.0 - D:/LAB PROJECT/NuSRS - 합친거/test.xml File Edit View Window Verification Help 🛛 🗠 🔍 🔍 🛄 Ê × 🔽 둘 🍍 8 🖹 Copy ⊖ Root × Paste 🗶 Delete ۰ ¥. SelectAll h_1 f_1 0 f 2 f_17 0 g_abc f3 h_3 f_4 **Description Window** ٠ 👇 📑 g_abc \bigcirc 🔶 🗂 Description ĿΝ 🔶 🚞 TemplateNumber \rightarrow LD × 🔶 📺 Input . 🗋 f_1 : boolean 🔁 Type Window ा 🔤 Console Window 🗋 f_2 : boolean 🗋 f_3 : boolean • f_1:boolean 🗋 f_4 : boolean f_17 : boolean ⊢ 📑 Output f_2 : boolean KAI5 Ŧ • 29

편집 > 붙여넣기 (1/4) 1. Edit > Paste 를 선택한다. (Ctrl + V 로도 가능) r 🛛 🖂 👕 NuSRS Editor 2.0 - D:/LAB PROJECT/NuSRS - 합친거/test.xml File Edit View Window Verification Help 👆 Cut 🛛 🗠 🔍 🔍 🔯 🗖 -{5 × 1 Ê 8 Copy ○ Root × W. 📄 Paste 🗶 Delete ٠ F. SelectAll f_1 h_1 0 f_2 f_17 \bigcirc g_abc f_3 h 3 f 4 Description Window 👇 📑 g_abc . \bigcirc - Call Description ĿΠ 🔶 📑 TemplateNumber \rightarrow - 🗅 🔶 🚞 Input 🗋 f_1 : boolean 🔁 Type Window 📴 Console Window f_2 : boolean f 3:boolean f 1:boolean ٠ = 🗋 f 4 : boolean f 17 : boolean 🔶 🚞 Output f 2:boolean KAI5 • 30

편집 > 붙여넣기 (2/4)

복사된 영역이 있으면 붙여 넣어지며, 이름은 기존의 이름들과 다른 이름으로 자동생성된다.

NuSRS Editor 2.0 - D:\LAB PROJECT\NuSR	S - 합친거test.xml	X
File Edit View Window Verification He	lp	
2 5 5 5		
Hierarchy Window		
<pre>Prime Root</pre>	$\begin{array}{c} \hline \\ \hline $	-
	$ \begin{array}{c} \hline f_{-4} \\ \hline h_{-1} \hline \hline h_{-1} \\ \hline h_{-1} \hline \hline h_{-1} \\ \hline h_{-1} \hline h_{-1} \\ \hline h_{-1} \hline \hline h_{-1} \\ \hline h_{-1} \hline \hline$	
— 🗋 f_1 : boolean		
— 🗋 f_2 : boolean	V lype window	
— 🗋 f_3 : boolean	f_1 : boolean	
🗆 🗋 f_4 : boolean	f_17 : boolean	Ξ
Output	f_17_: boolean	-



편집 > 붙여넣기 (3/4)

불여넣기 한 노드를 선택하고 마우스 오른쪽을 클릭하여 Rename을 선택한다.

NuSRS Editor 2.0 - D:/LAB PROJECT/NuSRS	- 합친거test.xml	a' X
File Edit View Window Verification Hel	p	
2 6 6 6		
Rierarchy Window	O Root ×	
P C Root □ g_abc	$\begin{array}{c} \hline \\ \hline $	
	$\begin{array}{c} \hline f_4 \\ \hline \hline f_1- \\ \hline \hline f_17_ \\ \hline \end{array}$	•
• Input		
- C f_2:boolean	🕼 Type Window 🧧 Console Window	
- 🗋 f_3 : boolean	f_1: boolean	
🗌 🗆 🗋 f_4 : boolean	f_17 : boolean	=
	f_17_: boolean	•



편집 > 붙여넣기 (4/4) 4. 붙여넣기 한 노드의 자동생성된 이름을 적절히 변경한다. ර් වේ 🛛 👕 NuSRS Editor 2.0 - D:\LAB PROJECT\NuSRS - 합친거\test.xml File Edit View Window Verification Help 🌮 🔒 × 🔞 📲 🍳 🔍 🔯 💶 💽 🍃 b ⊖ Root × 📲 Hierarchy Window ← ☐ Root F. └ 🗋 g_abc h_1 f_1 0 f 17 f_2 0 g_abc f_3 h3 f_4 Description Window h_5 P− □ Root \bigcirc 🔶 🚞 Description - 🗅 🔶 📑 TemplateNumber \rightarrow - D 🔶 🚞 Input • 🗋 f_1 : boolean 🔁 Type Window 🔪 🧰 Console Window ↑ f_2 : boolean 🗋 f_3 : boolean f_1 : boolean 🗋 f_4 : boolean f_17 : boolean 🔶 📺 Output f_17_: boolean KAIS 4 Ŧ • 33



편집 > 삭제 (2/3)

2. Edit > Delete를 선택한다.(Delete 키로도 가능)



KAIST

35


편집 > 모두 선택 (1/2) Edit > Select All을 선택한다.(Ctrl + A로도 1. 가능) 🧮 NuSRS Editor 2.0 - D:LAB PROJECT NuSRS - 합친거 test.xml r 🛛 🖂 File Edit View Window Verification Help 🗌 🚳 📲 🔍 🔍 🖾 🗖 👆 Cut Ê × 🔽 둘 🌋 -0 📄 Copy ○ Root × Paste 样 Delete R. SelectAll f_1 0 f_2 0 g_abc f_3 h 3 f_4 **Description Window** \bigcirc - C Description - 🗅 📹 TemplateNumber \rightarrow LΒ – 🗂 Input • 🗋 f_1 : boolean 🔁 Type Window 🧰 Console Window 🗋 f_2 : boolean 🗋 f_3 : boolean ۰ f_1 : boolean 🗋 f_4 : boolean f_2 : boolean 🔶 📺 Output f_3:boolean KAE 4 * 37



보기 > 찾기 (1/3)

1. View > Find를 선택한다.









보기 > I/O Variable 보기 (1/2)

1. View > View IO Vars를 선택한다.





보기 > I/O Variable 보기 (2/2)

2. I/O Variable의 list가 나타난다.

NuSRS Editor 2.0 - D:\LAB PROJECT\NuSRS	- 합친거itest.xml
File Edit View Window Verification Help	0
2 5 5 5 6	
📲 Hierarchy Window 🔹 🗖	
P 🚍 Root	
	🛃 I/O variables at Root 🛛 🗙
	// Input variables
	11 h_1
	f_4
	// Output variables h 3
Description Window	h_3 th_5
P→ C Description	
P- 🚍 Input	
-) f_1 : boolean	🙄 Type Window 🗧 Console Window
f_3 : boolean	f 1: boolean
_ 🗋 f_4 : boolean	f_17 : boolean
	f_2:boolean

보기 > 확대 (1/2)

View > Zoom In을 선택한다. *키 패드의 + 키로도 가능하다.





보기 > 확대 (2/2)

2. 현재 활성화되어 있는 창의 그림이 확대된다.



KAIST

보기 > 축소 (1/2)

1. View > Zoom Out을 선택한다.

* 키 패드의 – 키로도 가능하다.





46



보기 > 선택영역 확대 (1/3)

1. 마우스 왼쪽 클릭 또는 drag & drop으로 확대하고자 하는 영역을 선택한다.





보기 > 선택영역 확대 (2/3)

KALE

2. View > Zoom Selected를 선택한다.



보기 > 선택영역 확대 (3/3)

현재 선택되어 있는 영역이 창 전체에 차도록 창의 그림이 확대된다.



50

보기 > 화면 크기에 맞춤 (1/2)

1. View > Fit to Window를 선택한다.





보기 > 화면 크기에 맞춤 (2/2)

현재 활성화되어 있는 창의 그림이 창의 크기에 맞게 조정된다.

NuSRS Editor 2.0 - D:'LAB PROJECT'NuSRS	- 합친거test.xml	"ø" 🛛
File Edit View Window Verification Hel		
) 🗷 🗔 💌 🔍 🔍 🖾 🛄 💽 🚬 🌦	
Hierarchy Window		
P-□ Root □ g_abc		
	0 <u>f_3</u> g_abc h_3	
Description Window	<u> </u>	
P → Root P → Description	<u>f_5</u>	
←		
f_1 : boolean		•
- 🖸 f_2 : boolean	Type Window Console Window	(
- Lì f_3: boolean	f_1: boolean	
Output	[1/:boolean f_2:boolean 	



보기 > 자동 배치 (1/2)

1. View > Auto Align을 선택한다.





보기 > 자동 배치 (2/2)

KAIST

현재 활성화되어 있는 창의 노드 및 전이들 위치가 자동 조정되어 배치된다.

NuSRS Editor 2.0 - D:\LAB PROJECT\NuSR	6 - 합친거test.xml	r⊠ ⊠
File Edit View Window Verification He	lp	
2 🖓 🛃 🐝 🗎 🕻		
Hierarchy Window	O Root ×	
P □ Root □ g_abc	$\begin{array}{c} \mathbf{k} \\ \mathbf{c} \\ $	h_1
		th_5
L f_1 : boolean	🔁 Type Window 🔲 Consola Mindow	
f_2:boolean		
⊢ ∐ f_3: boolean	f_1 : boolean	^
t_4:boolean	f_17 : boolean	
	f_2 : boolean	



1. Window > Close 를 선택한다. 현재 활성화되어 있는 창이 닫힌다.

NuSRS Editor 2.0 - D:\LAB PROJECT\NuSRS	- 합친거ttest_3.xml 🔤 🖞	a' 🖂
File Edit View Window Verification Hel	þ	
Close All		
Hierarchy Window	○ Root × ○ th_5 ○ h_1 ○ f_17	
P-□ Root □ th_5 □ h_1 □ f_17 	$\begin{array}{c} \textbf{k} \\ \textbf{c} \\ $]
		•
- n f 4 : boolean	🙄 Type Window 🧧 Console Window	
- f_2:boolean	f 1: boolean	-
f_1 : boolean	f_17 : boolean	=
Output	f_2:boolean	-



모든 창 닫기 (1/2) 1. Window > Close All 을 선택한다. r 🛛 🖂 📰 NuSRS Editor 2.0 - D:\LAB PROJECT\NuSRS - 합친거\test_3.xml File Edit View Window Verification Help Kal 💶 🔽 Close 2 Ê 💌 둘 🗶 × 1 Close All 🔍 Hierarchy window ○ Root × ○ th_5 ○ h_1 ○ f_17 . ← ☐ Root K 🗋 th_5 🗋 h_1 th_5 th_5 🗆 🗋 f_17. f_3 h_1 Ο f_4 f_17 0 h_1 f_2 f_1 Description Window ۰ ← □ Root \bigcirc - C Description LΒ f_17 🔶 📑 TemplateNumber \rightarrow $-\square$ • 🔶 🚞 Input Ш ъI 🗋 f_3 : boolean 🔁 Type Window Console Window 🗋 f_4 : boolean f_2 : boolean * f 1:boolean └ 🗋 f_1 : boolean f_17 : boolean 🔶 🗂 Output f 2:boolean KAE Image: A start start

56

모든 창 닫기 (2/2)

2. 현재 열려져 있는 모든 창이 닫힌다.

	I 🔲 NuSRS Editor 2.0 - D:LAB PROJECT\NuSRS - 합친거\test_3.xml	r ⊠
	File Edit View Window Verification Help	
	Hierarchy Window	
	P	
	E Description Window	
KAIST	Image: Type Window Image: Console Window Image: Type Window I	

















Description Window



151

Software Engineering Lab. Deptpartment of Computer Science



Description (1/9) Maximize ▶현재 활성화되어 있는 Desctiprion 창에 대한 상세 정보를 **____** th_5 Description 보여준다. 🗋 th_5 노드에 대한 설명 삽입 🔶 📺 TemplateNumber ◆ FOD 이름 🔶 🗂 Input **f** 2 Description 윈도우는 현재 🗋 th 1 🗋 h 3 활성화되어 있는 창의 🔶 🗂 Output 🗋 th 5 이름(노드 이름)을 root로 🔶 🗂 Memoriazble Variable of External input **f_2_t0** : {10,20,40} 가진다. 🔶 🗂 Constant 🗋 k 1 := 4 ◆ Maximize 버튼 🔶 🗂 Local Clock Variable 🗋 time_1 : 0..4 🔶 🗂 Previous State Variable \mid 🗋 prev : boolean



Description (2/9)

E Desctiprion
P- □ th_5
🛉 🗂 Description
🗌 🗋 th_5 노드에 대한 설명 삽입
→ → → → → → → → → → → → →
r- 🚍 Input
- 🗋 f_2
- 🗋 th_1
h_3
P→ C= Output
🗌 🗆 🗋 th_5
🛉 🗂 Memoriazble Variable of External input
🗕 🗋 f_2_t0 : {10,20,40}
P→ Constant
└─ 🗋 k_1 := 4
🗣 🚍 Local Clock Variable
🗌 🗆 🗋 time_1 : 04
👇 🗂 Previous State Variable
🗕 🗋 prev : boolean

Description

- 현재 활성화된 다이어그램에 대한 설명을 기록한다.
- 종이 모양 아이콘에 대고 마우스 왼쪽 버튼을 더블클릭하면 편집할 수 있다.
- ▶ (FOD,SDT,FSM,TTS 모두 해당)



Description (3/9)



Template Number

- FSM,TTS로 표현되는 논리는 일정한 형식의 template을 가지고 있다. 해당 template number를 입력해 준다.
- 종이 모양 아이콘에 대고 마우스 왼쪽 버튼을 더블클릭하면 편집할 수 있다.
- ▶ (FSM, TTS만 해당)



Description (4/9)



Input

- 현재 활성화된 창의 입력변수를 모두 보여준다.
- Description 창에서 직접 수정은 불가능하다.
- 최상위 FOD에서 입력변수 이름이 변경된 경우, 하위 노드에서의 해당 입력 변수이름도 자동으로 변경된다.
- ▶ (FOD,SDT,FSM,TTS 모두 해당)



Description (5/9)

E Desctiprion	
- □ th_5	
- Carl Description	
→ → → → → → → → → → → → →	
• 🗂 Input	
$-\mathbf{D}f^2$	
\square \square \square \square	
\square \square \square th 5	
Gameriazhle Variable of External	Linnut
	mpar
$\square \square $	
Urrie_1 . U4	
Previous State Variable	
🗆 🗋 prev : poolean	



Output

- 현재 활성화된 창의 출력변수를 보여준다.
- Description 창에서 직접 수정은 불가능하다.
- 최상위 FOD에서 출력변수 이름이 변경된 경우, 하위 노드에서의 해당 출력 변수이름도 자동으로 변경된다.
- ▶ f_, h_, th_ 노드들은 Output 변수가 하나이다.
- ▶ (FOD,SDT,FSM,TTS 모두 해당) 68

Description (6/9)

E Desctiprion
♀- ☐ th_5 ♦- ☐ Description ↓ 10 th_5 노드에 대한 설명 삽입
TemplateNumber
f_2 th_1
← n3
Memoriazble Variable of External input f_2_t0 : {10,20,40}
- Constant - C k_1 := 4 - Cal Clock Variable
← ☐ time_1 : 04 ← ☐ Previous State Variable



Memorizable variable of External input

- 입력변수의 이전 주기값을 가리키는 변수를 선언한다.
- 입력변수명 뒤에 _t0, _t1 과 같은 접미어를 붙여서 몇 주기 전 값인지를 표현한다.
 - ▶t0는 한 주기 전, t1은 두 주기 전 등
- Memorizable variable of External input를 선택, 마우스 오른쪽 클릭하면 변수를 Add, Delete, Edit 할 수 있다.
- ▶ (FSM,TTS만 해당)

Description (7/9)

E Desctiprion
← 급 th_5 ← 급 Description └ ⓑ th_5 노드에 대한 설명 삽입 ← 급 TemplateNumber └ ⓑ ← 급 Input ← 읍 f_2 ← 읍 th_1 └ ⓑ h_3 ← 급 Output └ ⓑ th_5 ← 읍 th_5
Memoriazble Variable of External input 1 10,20,40 1 Constant 1 1 = 4
Local Clock Variable Image: Control Image: Control <

Constant

- 해당 다이어그램에서 사용되는 상수값을 선언한다.
- Constant를 선택, 마우스 오른쪽 클릭하면 상수를 Add, Delete, Edit 할 수 있다.
- ▶ (FOD,SDT,FSM,TTS 모두 해당)



Description (8/9)

E Desctiprion
P- □ th_5
🔶 🗂 Description
🗌 🗆 🗋 th_5 노드에 대한 설명 삽입
🔶 🗂 TemplateNumber
👇 🚍 Input
- 🗋 f_2
— 🗋 th_1
h_3
👇 🚍 Output
📙 🗋 th_5
🛉 🚽 Memoriazble Variable of External input
📙 🗋 f_2_t0 : {10,20,40}
👇 🗂 Constant
└─ 🗋 k_1 := 4
🛉 🗂 Local Clock Variable
📙 🖵 🗋 time_1 : 04
- Previous State Variable
🗌 🗆 🗋 prev : boolean

Local Clock Variable

- Transition label에 시간에 관한 조건을 사용하는
 TTS의 경우, Clock
 Variable을 선언해 주어야 한다.
- Local Clock Variable을 선택, 마우스 오른쪽 클릭하면 변수를 Add, Delete, Edit 할 수 있다.

▶ (TTS만 해당)



Description (9/9)



Previous State Variable

- 출력변수의 이전 주기값을 가리키는 변수를 선언한다.
- 출력변수명 뒤에 _t0, _t1 과 같은 접미어를 붙여서 몇 주기 전 값인지를 표현한다.

▶t0는 한 주기 전, t1은 두 주기 전 등

- Previous State Variable을 선택, 마우스 오른쪽 클릭하면 변수를 Add, Delete, Edit 할 수 있다.
- ▶ (FSM,TTS만 해당)
FOD,FSM(TTS),SDT 편집창



15

Software Engineering Lab. Deptpartment of Computer Science





FOD 편집창 – 세부 항목



FOD 편집창 – 버튼 (1/9)

Select

- ▶ 각 노드 및 transition을 선택할 때 사용한다.
- Selection 버튼을 누른 후, 편집창에서 선택하고자 하는 노드 및 transition 위치에서 마우스 왼쪽버튼을 클릭한다.
- Multiple Selection을 위한 마우스 drag&drop이 가능하다.



FOD 편집창 – 버튼 (2/9)

IO node

- 입출력 노드를 도식할 때 사용한다.
- IO node는 나가는 transition이나 들어오는 transition 중 하나에만 연결될 수 있다.
- IO node는 최상위 FOD에서만 생성될 수 있으며, 최상위 FOD에서 생성되었을 때 하위 FOD들에는 일관성 있게 자동으로 생성된다.
- IO node를 하위 FOD에서 지우면, 상위 FOD에서는 해당 transition이 삭제되고, IO node자체는 삭제되지 않는다.



FOD 편집창 – 버튼 (3/9)

IO node (cont.)

- IO node 버튼을 클릭하고 편집창에서 마우스 왼쪽버튼을 클릭한다.
- 노드 이름을 입력하는 대화창이 나오면, 노드 이름을 입력하고 확인을 클릭한다.
- 3. 노드의 이름이 입력됨과 동시에 노드가 도식된다.



FOD 편집창 – 버튼 (4/9)

• Group node

- g_로 시작되는 그룹 노드를 도식할 때 사용한다.
- 이름이 g_로 시작하지 않는 경우에는 에러 메시지가 출력되고 이름을 다시 입력 받는다.
- Group node 버튼을 클릭하고 편집창에서 마우스 왼쪽버튼을 클릭한다.
- 노드 이름을 입력하는 대화창이 나오면, 노드 이름을 입력하고 확인을 클릭한다.
- 3. 노드의 이름이 입력됨과 동시에 노드가 도식된다.



FOD 편집창 – 버튼 (5/9)

Function node

- f_로 시작되는 function 노드를 도식할 때 사용한다.
- 이름이 f_로 시작하지 않는 경우에는 에러 메시지가 출력되고 이름을 다시 입력 받는다.
- Function node 버튼을 클릭하고 편집창에서 마우스 왼쪽버튼을 클릭한다.
- 노드 이름을 입력하는 대화창이 나오면, 노드 이름을 입력하고 확인을 클릭한다.
- 3. 노드의 이름이 입력됨과 동시에 노드가 도식된다.



 \bigcirc

FOD 편집창 – 버튼 (6/9)

History node

- ▶ History node를 도식할 때 사용한다.
- 이름이 h_로 시작하지 않는 경우에는 에러 메시지가 출력되고 이름을 다시 입력 받는다.
- History node 버튼을 클릭하고 편집창에서 마우스 왼쪽버튼을 클릭한다.
- 노드 이름을 입력하는 대화창이 나오면, 노드 이름을 입력하고 확인을 클릭한다.
- 3. 노드의 이름이 입력됨과 동시에 노드가 도식된다.



FOD 편집창 – 버튼 (7/9)

Timed History node

- Timed History node를 도식할 때 사용
- 이름이 th_로시작하지 않는 경우에는 에러 메시지가 출력되고 이름을 다시 입력 받는다.
- Timed History node 버튼을 클릭하고 편집창에서 마우스 왼쪽버튼을 클릭한다.
- 노드 이름을 입력하는 대화창이 나오면, 노드 이름을 입력하고 확인을 클릭한다.
- 3. 노드의 이름이 입력됨과 동시에 노드가 도식된다.



 \bigcirc

FOD 편집창 – 버튼 (8/9)

Transition

- ▶ 노드간 transition을 도식할 때 사용한다.
- ▶ FOD에서 Self-Transition은 허용되지 않는다.
- Transition 버튼을 한 번 클릭한 후 다른 버튼을 누르기 전까지, 여러 개의 Transition을 그릴 수 있다.
- IO node는 그와 연결된 transition이 그려지는 순간 input node인지 output node인지 결정이 되며, 이 정보는 Description창의 Input/Output 부분에 바로 반영된다.



FOD 편집창 – 버튼 (9/9)

Transition (cont.)

- 1. Transition 버튼을 클릭한다.
- 2. Source 노드를 클릭하고 drag해서, Destination 노드에서 drop한다.
 - Group node에서 Group node로의 transition을 그릴 때는 transition이름을 입력해야 한다.
 - transition 이름은 source group node의 output node 이름 중 하나여야 한다.
 - Image of the second system of the second syste
 - transition 이름이 source group node의 output node 이름 중 하나가 아닌 경우는 source group node에 transition과 같은 이름의 output node가 자동으로 생성된다.



 \rightarrow

FOD 편집창 – 노드 이름 변경

1. 노드를 선택, 오른쪽 클릭하고 Rename을 선택한다.

🔲 NuSRS Editor 2.0 - D:LAB PROJECT/NuSRS - 합친거/test.xml	- d 🛛
File Edit View Window Verification Help	
2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
Hierarchy Window	
Root g_abc [] <th>h_1 f_17 h_3 th_5 f_5</th>	h_1 f_17 h_3 th_5 f_5

FOD 편집창 – 노드 이름 변경

 노드 이름이 편집할 수 있는 상태로 바뀌면 노드 이름을 변경하고, 마우스 왼쪽을 클릭해 주면 노드 이름 변경이 완료된다.





86

SDT 편집창 – 전체 화면

◆ f_로 시작되는 노드를 더블 클릭했을 경우 ◆ SDT를 도식할 수 있는 테이블 형태의 창

Nusks Earor 2.0 - D:LAB PROJECTNus	RS - 압신 기대St_J.XMI					3 2
File Edit View Window Verification F						
📲 Hierarchy Window		7 ×				
P ■ Root	Structure	Decision Table:				
		Conditions	1	2	3	
	h_3 > t.	_1				
						-
		Action	1	2	3	
	f_6 :=					
	f_6 :=					1
Description Window						_
f_17 f_17 f_17						
 TemplateNumber 						
🕈 🚍 Input						
f_3 : boolean						
← ☐ Output		III.				-
└─ [] f_17 : boolean • □ Constant	Type Window	Console Window				
true := 1	f 17 : boolean					_
L Clock Variable	f_2 : boolean					
- Previous State Variable	f_3 : boolean					
└ 🗋 Memoriazble Variable of External i	input					



	Root O f_17 ×					
unction	node 이름 Structured Decision Tal	Close				
	h_3 > th_1	Conditions	1	2	3	
						SC
	f_6 := 1	Action	1	2	3	
	f_6 := 0					

SDT 편집창 – 세부 항목 (2/2)



Structured Decision Table

- ▶ NuSCR의 semantics에 의거하여 구성된 table
- 각 행과 열에서 오른쪽 마우스를 클릭하면 행, 열의 추가와 삭제, 복사, 붙여넣기, 지우기 등이 가능하다.





FSM, TTS 편집창 – 세부 항목





FSM, TTS 편집창 – 버튼 (1/3)

Select

- State 및 transition을 선택할 때 사용한다.
- Selection 버튼을 누른 후, 편집창에서 선택하고자 하는 State 및 transition 위치에서 마우스 왼쪽버튼을 클릭한다.
- Multiple Selection을 위한 마우스 drag&drop이 가능하다.



FSM, TTS 편집창 – 버튼 (2/3)

State

0

- ▶ State를 그릴 때 사용한다.
- State 이름에는 공백이나 특수문자가 포함되어서는 안 된다.
- State 버튼을 클릭하고 편집창에서 마우스 왼쪽버튼을 클릭한다.
- State 이름을 입력하는 대화창이 나오면, State 이름을 입력하고 확인을 클릭한다.
 - State 이름을 잘못 입력한 경우에는 에러 메시지가 출력되고 이름을 다시 입력 받는다.
- 3. State의 이름이 입력됨과 동시에 state가 도식된다.



FSM, TTS 편집창 – 버튼 (3/3)

Transition

- State간 transition을 도식할 때 사용한다.
- FSM 및 TTS에서는 최대 한 개까지만 Self-Transition을 그릴 수 있다.
- Transition 버튼을 한 번 클릭한 후 다른 버튼을 누르기 전까지, 여러 개의 Transition을 그릴 수 있다.
- 1. Transition 버튼을 클릭한다.
- 2. Source state를 클릭하고 drag해서, Target state에서 drop한다.



FSM, TTS 편집창 – 상태 이름 변경 (1/2)

1. 상태를 선택, 오른쪽 클릭하고 Rename을 선택한다.



95

FSM, TTS 편집창 – 상태 이름 변경 (2/2)

 상태 이름이 편집할 수 있는 상태로 바뀌면 상태 이름을 변경하고, 마우스 왼쪽을 클릭해 주면 노드 이름 변경이 완료된다.

NuSRS Editor 2.0 - D:\LAB PROJECT\NuSRS	- 합친거test_3.xml 👘 🖉
File Edit View Window Verification Help	
2 5 5 4 2 2 2	
Reference window	□ O Root O th_5 × [
P-□ Root	
- D h_1	
└─ 🖸 f_17	0
	\rightarrow f_2_t0 > f_2 & th_1 >= h_3 / th_5 := 1
	State 2
	(k 1. k 1) prev
E Description Window	State_1
₽ th_5	
P→ □ Input	
f_4 : boolean	
P-	Vindow Console Window
L D th_5: boolean	f_3:boolean
Constant	th 5 · hoolean

96

KAI

FSM, TTS 편집창 – 초기 상태 변경 (1/2)

상태를 선택, 오른쪽 클릭하고 Set Initial State를 선택한다.

Hierarchy Window Hierarchy Window Road Itics Road Itics <th>NuSRS Editor 2.0 - D:\LAB PROJEC</th> <th>NuSRS - 합친거ttest_3.xml</th> <th>r d X</th>	NuSRS Editor 2.0 - D:\LAB PROJEC	NuSRS - 합친거ttest_3.xml	r d X
Herarchy Window	P 😜 📄 🍕 😽 🛙	• • • • • • • • • • • • • • • • • • •	*
Root h_1 f_17 f_17 f_17 f_17 f_17 f_17 f_17 f_117 <p< th=""><th>📲 Hierarchy Window</th><th>Root O Root O th_5 ×</th><th></th></p<>	📲 Hierarchy Window	Root O Root O th_5 ×	
Description Window Image: Console Window Image: Constant <	P - Root h_5 h_1 f_17		f_2_10 > f_2 & th_1 >= h_3 / th_5 := 1
<pre>th_5 Description TemplateNumber Input 1_3: boolean f_4: boolean f_3: boolean f_4: boolean f_4: boolean f_4: boolean f_4: boolean </pre>	Description Window	(k_1, k_1) prev <= th	54Th_5:=0
Console and Console Window Console Win			
Constant Type vintuov Type vin	f_4 : boolean		
	← 📑 Output	f 3: boolean	
- 🗋 true := 1 🗾 🚽 th_5 : boolean	⊷ 📑 Constant	f_4 : boolean	
	- 🗋 true := 1	▼ th_5 : boolean	



FSM, TTS 편집창 – 초기 상태 변경 (2/2)

선택한 상태가 초기 상태로 바뀌고, 기존의 초기 상태는 보통 상태로 돌아간다.





98



Type Window

◆ 활성화된 다이어그램에 사용된 모든 변수들을 기록하고, 각 변수의 타입을 편집할 수 있는 창





100



1. 변수를 선택, 더블클릭한다.





Type Window 편집 (2/3)

변수 타입 입력창에서 변수의 타입을 선택 및 입력하고 OK를 누른다.

f_1 🔀	
name : f_1	기본 설정인 boolean에서
O Value	Range 타입으로 변경
🔾 Boolean	
Range Start: 0 End: 100	
Cancel Add Remove	







◆ 변수 이름이 중복되지 않도록 체크





검증 > Quick Check (1/4)

1. Verification > Quick Check을 선택한다.





검증 > Quick Check (2/4)
2. Type Window 우측의 Check 결과를 확인한다.
Interactive Window Verification Heip Image: Section 2.0-c:Documents and Settings/GRS:HEIP #1940bisR5.2007120269P.20051019-Final-Verification.xml Image: Section 2.0-p/2R_PRESS x Image: Section 2.0-p/2R_PRESS Image: Section 2.0-p/2R_PRESS x Image: Section 2.0-p/2R_PRESS x Image: Section 2.0-p/2R_PRESS _ Int_SP Image: Section 2.0-p/2R_PRESS x Image: Section 2.0-p/2R_PRESS x Image: Section 2.0-p/2R_PRESS _ Int_SP, Transition Rising -> Falling, Condition - Undefined Variable : f_LO_PZR_PRESS_Val_Ou_t1t Error At f_LO_PZR_PRESS_Int_SP, Action Row : 0 Col : 0 - Undefined Variable : f_LO_PZR_PRESS_Int_SP
Imput Imput Im

검증 > Quick Check (3/4)

◆Quick Check는 현재 편집중인 FOD, SDT 혹은 FSM/TTS와 그 하위 요소까지 모두 체크한다.

◆Root 노드의 편집 중에 Quick Check를 실행하면 전체 모델에 대해 Check를 수행한다.


검증 > Quick Check (4/4)

- ◆ Quick Check에서 Check되는 항목들은 다음과 같다.
 - 1. SDT Condition, Action의 Undefined Variable
 - 2. SDT Condition, Action에서 연산자 사용 잘못
 - 3. SDT에서 한 개의 Condition에 둘 이상의 Action 할당
 - 4. FSM, TTS Transition의 Undefined Variable
 - 5. FSM, TTS Transition의 연산자 사용 잘못
 - 6. FOD, FSM, TTS에서 Transition이 없는 노드
 - 7. FSM, TTS에서 Initial State로부터 Unrechable 노드
 - 8. FOD에서 Output 변수와 연결된 노드 동일 명칭 체크



검증 > 모델체킹 (1/6)

1. Verification > Model Checking을 선택한다.



검증 > 모델체킹 (2/6) 현재 활성화된 창의 NuSCR 명세를 2. SMV입력언어로 변환한 결과 창이 뜬다. In NuSRS Editor 2.0 - D:\LAB PROJECT\NuSRS - 합친거\BP-20051019-Final-Verification.xml File Edit View Window Verification Help 💿 🚛 🔲 🚛 🔽 Quick Check 🔍 🔍 🔍 🚺 🗖 Ś 1 ≜ C:₩Documents and Settings₩지은경₩My Documents₩과제₩KNICS 프로젝... 🗙 el Checking O Root O g_BP O g_HI_LOG_POWER × SMV f_Mod_Err:m_f_Mod_Err; R 1_HI_LOG_POWER_Opb_RopH_MGR THE OG_POWER_OP +>LOG_POWER_Op_Byp_Init f_HI_LOG_POWER_Opb_Rqst_MCR : m_f_HI_LOG_POWER_Opb_Rqst_MCR; 1_HI_LOG_POWER_Opb_Rost_RSF f HI LOG POWER Opb Rgst RSR: m f HI LOG POWER Opb Rgst RSR; f_HI_LOG_POWER_Op_Byp_Init:m_f_HI_LOG_POWER_Op_Byp_Init(f_HI_LOG_POWER_Opb_Rqst_f 1_H_UOG_POWER_Opb_Pam_k LOG POWER OB En f_HI_LOG_POWER_OB_Err:m_f_HI_LOG_POWER_OB_Err(f_HI_LOG_POWER_Opb_Rqst_MCR.f_HI LOG POWER f_HI_LOG_POWER_Opb_Perm_In : m_f_HI_LOG_POWER_Opb_Perm_In; f_HI_LOG_POWER_Chan_Err: m_f_HI_LOG_POWER_Chan_Err; \circ HI_LOG_POWER_PV cycle : 0..20; 1 HI LOG POWER MT VA LOG_POWER H_LOG_POWER_Val_Out 1 HI LOG POWER AT VA ASSIGN \bigcirc nit(cycle) :=0; 1 HI LOG POWER P HI (LOG POWER next(cycle) := case LOG POWER Ptrp 1 H LOG_POWER_MT/Dug cycle < 20 : cycle + 1; 1_H LOG_POWER_PT_Query cycle = 20:0;LOG POWER Trip Logi th_HILOG_POWER f_HLLOG_POWER_AT_Query \bigcirc ILOG_POWER_PV_Er 1 HILOG POWER P Save Close Execution PROPERTY LOG_POWER LOG_POWER_Ptp_Out 정/상승/ f Mod Err 1 HI LOG POWER Char Er H LOG POWER LOG_POWER_Trip_Out boolear Val : bo 🎲 Type Window 🛛 🔤 Console Window Val : boo f_HI_LOG_POWER_AT_Query : boolean . Val : bo Apply f_HI_LOG_POWER_AT_Val : boolean f HI LOG POWER Chan Err:boolean f_HI_LOG_POWER_PT_Query: • f_HI_LOG_POWER_MT_Query : boolean KAIS 111

검증 > 모델체킹 (3/6)

3. 검증속성을 입력 후, Apply 버튼을 누르면 SMV 입력에 반영된다. (SMV 입력창 편집 가능)

SMV	
init(cycle) :=0:	
unitation - of	
next(cycle) := case	
cycle < 20 : cycle + 1;	
cycle = 20 : 0;	
1 : cycle;	
esac,	
DEFINE	
sec := cycle = 20;	
true:= 1;	
false:= 0;	
Save Close Execution PROPERTY	
 AG(f_HI_LOG_POWER_Op_Byp_Init.f_HI_LOG_POWER_Op_Byp_Init=true & (f_Mod_Err.f_Mod_Err.	᠊᠊᠊᠊᠊᠊᠊᠋᠊᠋᠊᠆᠋᠋᠊᠋᠋᠊᠋᠊᠋᠊᠋
	(CTL 형신)
Apply	



검증 > 모델체킹 (4/6)

4. Execution버튼을 누르면 SMV 도구가 실행된다.

囊 C:₩Documents and Settings₩시은경₩My Documents₩과제₩KNICS 프로젝 🔀	<u>File Prop View Goto Hi</u> story <u>A</u> bstraction
⊂SMV	
	<u>B</u> rowser Pr <u>o</u> perties <u>R</u> esults <u>C</u> one Usi <u>ng</u> <u>G</u> roups
init(cycle) :=0; —	
next(cycle) := case	Name Layer
cycle < 20 : cycle + 1;	⊞ <u>(top level)</u>)
cycle = 20 : 0;	
1 : cycle;	
esac;	
DEFINE	Source Trace Log
sec := cycle = 20;	
true:= 1;	Fil <u>e</u> Sho <u>w</u>
ifalse:= 0;	
	FROM-Normal-TO-Waiting-taken : Waiting;
SPEC AG(f_HI_LOG_POWER_Op_Byp_Init.f_HI_LOG_POWER_Op_Byp_Init= true & (f_Mod_Err.f_Mod_E	I : SIAIL;
	esac;
	Outputs
	init(th HI LOG POWER Ptrn Logic) := 0:
Save Close Execution	next (th HI LOG POWER Ptrn Logic) := case
	FROM-Waiting-TO-Normal-taken : false:
PROPERTY	FROM-Waiting-TO-Pretrin-taken : true:
	FROM-Pretrip-TO-Normal-taken : false;
AG(f_HI_LOG_POWER_Op_Byp_Init.f_HI_LOG_POWER_Op_Byp_Init = true & (f_Mod_Err.f_Mod_Err = true	1 : th HI LOG POWER Ptrp Logic;
	esac;
	Increment local clock time 1
	init(time 1) := 0;
	next(time 1) := case
	sec & in-Waiting & time 1 < k HI LOG POWER Ptrp Dly & (
	f HI LOG POWER Val Out >
Apply	k_HI_LOG_POWER_Ptrp_Set) : time_1 + 1;
	1:0;
	esac;



113

Help

검증 > 모델체킹 (5/6)

5. SMV에서 Prop > Verify 또는 Verify all을 선택한다.

74 defa	74 default.smy								
<u>F</u> ile <u>P</u>	rop <u>V</u> iew	<u>G</u> oto	H <u>i</u> story	<u>A</u> bstrac	tion		_		<u>H</u> elp
<u>B</u> ro' [−]	Options Copy optio Group sibli Verify . Verify <u>a</u> ll State coun	ns ngs t	<u>R</u> esults	<u>C</u> one	Usi <u>ng</u>	<u>G</u> roup	is		
Sourc	e <u>T</u> race	Log							
Fil <u>e</u>	Fil <u>e</u> Sho <u>w</u>								
FROI 1 : esac; Out	<pre>FROM-Normal-TO-Waiting-taken : Waiting; 1 : STATE; esac; Outnuts</pre>								
<pre>init(th_HI_LOG_POWER_Ptrp_Logic) := 0; inext(th_HI_LOG_POWER_Ptrp_Logic) := case FROM-Waiting-TO-Normal-taken : false; FROM-Waiting-TO-Pretrip-taken : true; FROM-Pretrip-TO-Normal-taken : false; 1 : th_HI_LOG_POWER_Ptrp_Logic; esac;</pre>									
<pre> Increment local clock time_1 init(time_1) := 0; next(time_1) := case sec & in-Waiting & time_1 < k_HI_LOG_POWER_Ptrp_Dly & (f_HI_LOG_POWER_Val_Out > k_HI_LOG_POWER_Ptrp_Set) : time_1 + 1; 1:0; esac;</pre>									



검증 > 모델체킹 (6/6)



74 default.smv 📃 🗖 🔀	74 default.smv 📃 🗆 🔀
<u>F</u> ile <u>P</u> rop <u>V</u> iew <u>G</u> oto History <u>A</u> bstraction <u>H</u> elp	<u>F</u> ile <u>P</u> rop ⊻iew <u>G</u> oto History <u>A</u> bstraction <u>H</u> elp
<u>B</u> rowser Properties <u>R</u> esults <u>C</u> one Usi <u>ng</u> <u>G</u> roups	<u>B</u> rowser Properties <u>R</u> esults <u>C</u> one Usi <u>ng</u> <u>G</u> roups
All results	All results
Property Result	Property Result
(AG (((f_HI_LOG_POWER_Op_Byp_Init.f_HI_LOG_POWER_Op_Byp_Init=1)&(((f_M_falseTue	(AG (((f_HI_LOG_POWER_Op_Byp_Init.f_HI_LOG_POWER_Op_Byp_Init=1)&(((f_M_falseTue
Source Trace Log	Source Trace Log
File	Fil <u>e</u> E <u>d</u> it R <u>u</u> n Vie <u>w</u>
Model checking time: 0.093750	3
user time0.1 40625 s	f HI LOG POWER PV Err.false 0
system time0.0	f HI LOG_POWER_PV_Err.\ininit0
13625 3	f_HI_LOG_POWER_PV_Err.\in-s0
Model checking results	f_HI_LOG_POWER_PV_Err.\in-s1 1
(AG (((f_HI_LOG_POWER_Op_Byp_Init.f_HI_LOG_POWER_Op_Byp_Init=1)&(((f_M	f_HI_LOG_POWER_PV_Err.true 1
false	f_HI_LOG_POWER_Trip_Out.\FROMinitTO-s0-enabled 0
See file "default.warn" for warnings. user time	f_HI_LOG_POWER_Trip_Out.\FROMinitTO-s0-taken 0
	f_HI_LOG_POWER_Trip_Out.\FROMinitTO-s1-enabled 0
	f_HI_LOG_POWER_Trip_Out.\FROMinitTO-s1-taken
	f_HI_LOG_POWER_Trip_Out.\FROM-s0-TO-s0-enabled
12072 2	

