CS 671 Automated Reasoning

Introduction to NuPRL

1. **NuPRL Features**
2. **NuPRL Architecture**
3. **Interactive Theorem Proving in NuPRL**
   - Decomposition & Computation
   - Defining new constructs

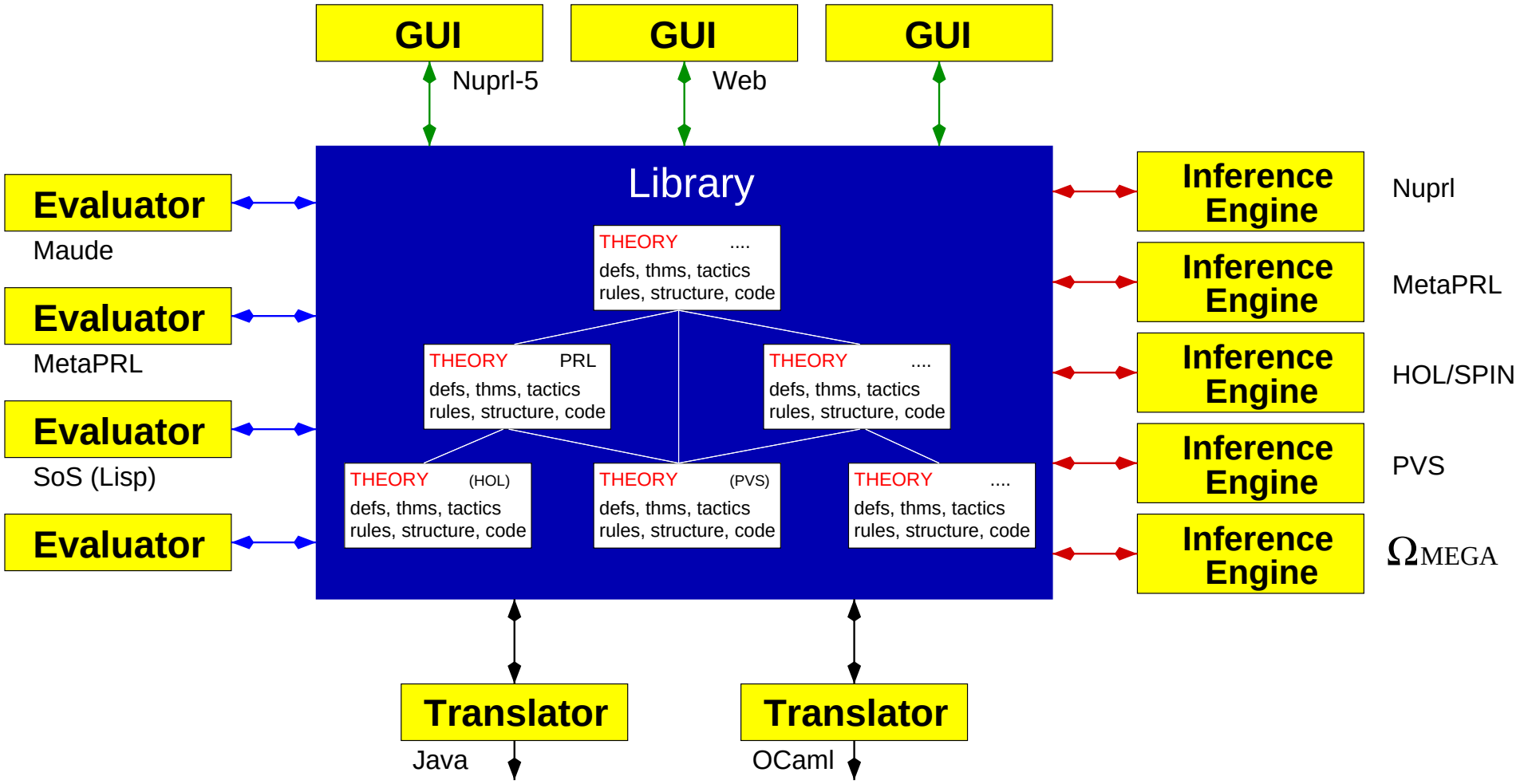
The Nuprl System

- **Beginnings in 1984**
  - Nuprl 1 (Symbolics): proof & program refinement in Type Theory
  - Book: *Implementing Mathematics ...* (1986)
  - Nuprl 2: Unix Version
- **Nuprl 3: Mathematical Problem Solving**
  - Machine proof for unsolved problems (Girard's paradox) (Howe 1987)
    (Higman's Lemma) (Murthy 1990)
- **Nuprl 4: System Verification and Optimization**
  - Verification of a logic synthesis tool (Aagaard & Leeser 1993)
  - Verification of the SCI cache coherency protocol (Howe 1996)
  - Optimization of the Ensemble group communication system (Kreitz, Hayden & Hickey 1999)
  - Verification of Ensemble protocol layers (Bickford 1999)

Nuprl 4 System Features

- **Interactive Proof Editor** ⇝ readable proofs
- **Flexible definition mechanism** ⇝ user-defined terms
- **Customizable Term Display** ⇝ flexible notation
- **Structure Editor for Terms** ⇝ no ambiguities
- **Tactics** ⇝ user-defined inferences
- **Decision Procedures**
- **Proof objects, Program Extraction** ⇝ program synthesis
- **Program Evaluation**
- **Library mechanism** ⇝ user-theories
  - Large mathematical libraries
  - Large tactics collection
- **HTML output generator** ⇝ web accessibility

Nuprl 5: An Open Logical Environment



Platform for Cooperating Reasoning Systems

Nuprl 5: Additional System Features

- **Collection of Cooperating Processes**
  - Centered around a common knowledge base
  - Refiners, interfaces, evaluators, etc. connect as independent processes
  - Processes can connect and disconnect at any time
- **Ability to Connect to External Systems**
  - MetaPRL, JProver, HOL, ...
- **Library Organized as Persistent Data Base**
  - Transaction model + Version control + Dependency tracking
- **Reflective System Structure**
  - System designed within the system's library $\rightsquigarrow$ customizable structure
- **Cooperating Inference Engines**
  - Asynchronous and distributed theorem proving
- **Multiple User Interfaces**
  - Structure editor, Web front end, ...

Refinement Rules for Natural Numbers

$H \vdash \mathbb{N}$ `type` — natR

$H \vdash$ `0=0` $\in \mathbb{N}$ — zeroR

$H \vdash$ `suc(`e`)=suc(`e'`)` $\in \mathbb{N}$ — sucR

$H \vdash e = e' \in \mathbb{N}$

$H \vdash$ `ind(`e`;` $base$`;` $n, x.up$`)` $=$ `ind(`e'`;` $base'$`;` $n', x'.up'$`)` $\in T$ — indR

$H \vdash e = e' \in \mathbb{N}$

$H \vdash base = base' \in T$

$H, n:\mathbb{N}, x:T \vdash up = up'[n, x / n', x'] \in T$

$H_1, x:T, H_2 \vdash x = x \in T$ — hypotheses

$H \vdash$ `ind(0;` $base$`;` $n, x.up$`)` $= e' \in T$ — compute 1

$H \vdash base = e' \in T$

$H \vdash$ `ind(suc(`e`);` $base$`;` $n, x.up$`)` $= e' \in T$ — compute 1

$H \vdash up[e,$ `ind(`e`;` $base$`;` $x.n, up$`)` $/ n, x] = e' \in T$

Refinement Rules for Function Spaces

$H \vdash S \rightarrow T$ `type` — `funR`

$H \vdash S$ `type`

$H \vdash T$ `type`

$H \vdash \lambda x.e = \lambda x'.e' \in S \rightarrow T$ — `lamR`

$H, x:S \vdash e = e'[x/x'] \in T$

$H \vdash S$ `type`

$H \vdash f\ e = f'\ e' \in T$ — `appR` $S \rightarrow T$

$H \vdash f = f' \in S \rightarrow T$

$H \vdash e = e' \in S$

$H \vdash (\lambda x.e)\ e' = e^* \in T$ — `compute 1`

$H \vdash e'[e/x] = e^* \in T$

Note: $e = e \in T$ is usually abbreviated by $e \in T$

Refinement Rules for First-Order Logic

	left	right	
`andL` i	$H,\ A \wedge B,\ H' \vdash G$ $H,\ A,\ B,\ H' \vdash G$	$H \vdash A \wedge B$ $H \vdash A$ $H \vdash B$	`andR`

`orL` i	$H,\ A \vee B,\ H' \vdash G$
	$H,\ A,\ H' \vdash G$
	$H,\ B,\ H' \vdash G$
		$H \vdash A \vee B$
		$H \vdash A$
		
		$H \vdash A \vee B$
		$H \vdash B$
			`orR1`
			
			`orR2`

`impL` i	$H,\ A \Rightarrow B,\ H' \vdash G$ $H,\ A \Rightarrow B,\ H' \vdash A$ $H,\ H',\ B \vdash G$	$H \vdash A \Rightarrow B$ $H,\ A \vdash B$	`impR`
`notL` i	$H,\ \neg A,\ H' \vdash G$ $H,\ \neg A,\ H' \vdash A$	$H \vdash \neg A$ $H,\ A \vdash$ `false`	`notR`
`exL` i	$H,\ \exists x{:}T\,.\,B,\ H' \vdash G$ $H,\ x{:}T,\ B,\ H' \vdash G$	$H \vdash \exists x{:}T\,.\,B$ $H \vdash B[t/x]$	`exR` t
`allL` i t	$H,\ \forall x{:}T\,.\,B,\ H' \vdash G$ $H,\ \forall x{:}T\,.\,B,\ B[t/x],\ H' \vdash G$	$H \vdash \forall x{:}T\,.\,B$ $H,\ x{:}T \vdash B$	`allR`

Note: an unlabelled hypotheses A is an abbreviation for `%:A`