CS 671 Automated Reasoning

Tactical Theorem Proving in NuPRL

1. Basic Tactics
2. Tacticals
3. Advanced Tactics
   Chaining, Induction, Case Analysis
Tactics: User-defined inference rules

- **Meta-level programs built using**
  - Basic inference rules
  - Predefined tacticals . . .
  - Meta-level analysis of the proof goal and its context
  - Large collection of standard tactics in the library

- **May produce incomplete proofs**
  \[\rightarrow\] User has to complete the proof by calling other tactics

- **May not terminate**
  \[\rightarrow\] User has to interrupt execution

but

Applying a tactic always results in a valid proof
Basic Tactics

Subsume primitive inferences under a common name

- **Hypothesis**: Prove \( \ldots C \ldots \vdash C' \) where \( C' \) \( \alpha \)-equal to \( C \)
- **Declaration**: Prove \( \ldots x:T \ldots \vdash x \in T' \) where \( T' \) \( \alpha \)-equal to \( T \)
  - Variants: NthHyp \( i \), NthDecl \( i \)

- **D \( c \)**: Decompose the outermost connective of clause \( c \)

- **EqD \( c \)**: Decompose immediate subterms of an equality in clause \( c \)
  - **MemD \( c \)**: Decompose subterm of a membership term in clause \( c \)
    - Variants: EqCD , EqHD \( i \), MemCD , MemHD \( i \)

- **EqTypeD \( c \)**: Decompose type subterm of an equality in clause \( c \)
  - **MemTypeD \( c \)**: Decompose type subterm of a membership term in clause \( c \)
    - Variants: EqTypeCD , EqTypeHD \( i \), MemTypeCD , MemTypeHD \( i \)

- **Assert \( t \)**: Assert (or cut) term \( t \) as last hypothesis

- **Auto**: Apply trivial reasoning, decomposition, decision procedures ...

- **Reduce \( c \)**: Reduce all primitive redices in clause \( c \)
TACTICALS

- \( \text{tac}_1 \ \text{THEN} \ \text{tac}_2 \): Apply \( \text{tac}_2 \) to all subgoals created by \( \text{tac}_1 \)
- \( \text{tac}_1 \ \text{THENL} \ [\text{tac}_1; \ldots; \text{tac}_n] \): Apply \( \text{tac}_i \) to the \( i \)-th subgoal created by \( \text{tac}_1 \)
- \( \text{tac}_1 \ \text{THENA} \ \text{tac}_2 \): Apply \( \text{tac}_2 \) to all auxiliary subgoals created by \( \text{tac}_1 \)
- \( \text{tac}_1 \ \text{THENW} \ \text{tac}_2 \): Apply \( \text{tac}_2 \) to all wf subgoals created by \( \text{tac}_1 \)

- \( \text{tac}_1 \ \text{ORELSE} \ \text{tac}_2 \): Apply \( \text{tac}_1 \). If this fails apply \( \text{tac}_2 \) instead

- \( \text{Try} \ \text{tac} \): Apply \( \text{tac} \). If this fails leave the proof unchanged

- \( \text{Complete} \ \text{tac} \): Apply \( \text{tac} \) only if this completes the proof

- \( \text{Progress} \ \text{tac} \): Apply \( \text{tac} \) only if that causes the goal to change

- \( \text{Repeat} \ \text{tac} \): Repeat \( \text{tac} \) until it fails
  - \( \text{RepeatFor} \ i \ \text{tac} \): Repeat \( \text{tac} \) exactly \( i \) times

- \( \text{AllHyps} \ \text{tac} \): Try to apply \( \text{tac} \) to all hypotheses
- \( \text{OnSomHyp} \ \text{tac} \): Apply \( \text{tac} \) to the first possible hypotheses
Supplying Parameters to Tactics

- Position of a hypothesis to be used: \text{NthHyp} \( i \)
- Names for newly created variables: \text{New} \( \{x\} \) (D 0)
- Type of some subterm in the goal: \text{With} \( x:S \rightarrow T \) (MemD 0)
- Term to instantiate a variable: \text{With} \( s \) (D 0)
- Universe level of a type: \text{At} \( j \) (D 0)
- Dependency of a term instance \( C[z] \) on a variable \( z \): \text{Using} \( [z,C] \) (D 0)
Advanced Tactics: (Inductive) Analysis

• Induction
  - NatInd \( i \): standard natural-number induction on hypothesis \( i \)
  - IntInd, NSubsetInd, ListInd: induction on \( \mathbb{Z}, \mathbb{N} \) subranges, lists
  - CompNatInd \( i \): complete natural-number induction on hypothesis \( i \)

• Case Analysis
  - BoolCases \( i \): case split over boolean variable in hypothesis \( i \)
  - Cases \([t_1;\ldots;t_n]\): \( n \)-way case split over terms \( t_i \)
  - Decide \( P \): case split over (decidable) proposition \( P \) and its negation
Advanced Tactics: Chaining

- **Instantiating Facts**
  - InstHyp \([t_1; \ldots; t_n]\) \(i\): instantiate hypothesis \(i\) with terms \(t_1 \ldots t_n\)
  - InstLemma \(name\) \([t_1; \ldots; t_n]\): instantiate lemma \(name\) with terms \(t_1 \ldots t_n\)

- **Forward Chaining**
  - FHyp \(i\) \([h_1; \ldots; h_n]\): forward chain through hypothesis \(i\) matching its antecedents against any of the hypotheses \(h_1 \ldots h_n\)
  - FLemma \(name\) \([h_1; \ldots; h_n]\): forward chain through lemma \(name\)

  Optional argument \(Sel\) \(n\)

- **Backward Chaining**
  - BHyp \(i\): backward chain through hypothesis \(i\) matching its consequent against the conclusion of the proof
  - BLemma \(name\): backward chain through lemma \(name\)
  - Backchain \(bc\_names\): backchain repeatedly through lemmas and hypotheses

  Optional argument \(Using\) \(binding\)
Running Nuprl from a Unix machine

Copy the file `nuprl/utils/profile/nuprl.config.cs671` to `~/.nuprl.config`
Edit `.nuprl.config` and change the entries
(iam "YourNameHere")
(sockets 1289 1980)
You may change the 0 to any number between 1-9. DO NOT change 1289!
In an xterm execute
xset fp+  nuprl/fonts/bdf
xset fp rehash
xhost +baldwin
rsh baldwin /usr/bin/X11/xterm -display ‘hostname’:0 -ls
Using baldwin makes sure that there are no memory issues. You may have to adjust the
-display setting. You also may want to add `~ nuprl/bin` to your path, e.g. by typing (in csh)
set path = ( `nuprl/bin $path`) into the new window.
On baldwin execute  nuprl/bin/emacs  nuprl
In emacs type (m-x)nuprl
This should run for a minute then pop up the Nuprl windows on the display.
In the navigator, go into the directories theories, then users, click MkTHY*, enter your name
into [token], click OK* and work only in the newly created theory
To quit, type stop. into the emacs shell after the ML[(ORB)]> prompt.