

Key components	Explanation
Name	Load Hedge Definition
Actors	User
Description	Load definition of the hedge algebra to be used
Goal	To generate the truth value domain of the knowledge base
Pre-condition	There exists the database containing hedge algebra's definition
Trigger	The system begins to execute
Post-condition	Logic system initialized with proper truth value domain.
Normal flow	[User executes the main program] 1. The system connects to the database 2. The hedges definition is loaded 3. Enters main loop and prints system's menu if successful
Alternative flow	
Exception flow	[Database connection error] <ul style="list-style-type: none"> • Inform the user • Exit

Key components	Explanation
Name	Load Knowledge Base
Actors	User
Description	Load KB into the system
Goal	To load KB
Pre-condition	Hedge definition loaded KB database exists
Trigger	User chose option Load KB
Post-condition	KB loaded
Normal flow	[User chose Load KB] 1. The system connects to the KB database 2. The system query the database for the clause table 3. If successful, trigger Check Consistency 3.1. If successful, load KB, trigger Show KB and return to main loop 3.2. Otherwise, drop KB, inform the user and return to main loop
Alternative flow	[<i>Improper format</i>] <ul style="list-style-type: none"> • Inform the user • Return to main loop
Exception flow	[<i>Null KB</i>] <ul style="list-style-type: none"> • Inform the user • Return to main loop [<i>Database connection error</i>] <ul style="list-style-type: none"> • Inform the user • Return to main loop

Key components	Explanation
Name	Show Knowledge Base
Actors	User
Description	Show the loaded KB
Goal	To show the loaded KB
Pre-condition	KB loaded
Trigger	User chose option Show KB
Post-condition	KB showed
Normal flow	[User chose Show KB] 1. KB printed.
Alternative flow	
Exception flow	[Null KB] <ul style="list-style-type: none"> • Inform the user • Return to main loop

Key components	Explanation
Name	Drop Knowledge Base
Actors	User
Description	Drop the KB
Goal	To drop the KB
Pre-condition	KB exists
Trigger	User chose option Drop KB
Post-condition	KB dropped
Normal flow	[User chose Drop KB] 1. KB dropped.
Alternative flow	
Exception flow	[Null KB] <ul style="list-style-type: none"> • Inform the user • Return to main loop

Key components	Explanation
Name	Write Knowledge Base
Actors	User
Description	Commit all changes in the loaded KB to the database
Goal	To permanently changed the KB database
Pre-condition	One or more queries made
Trigger	User chose option Write KB
Post-condition	KB written
Normal flow	[User chose Write KB] 1. The system connect to the KB database 2. The system commit all the queries user has made 3. If successful, return to main loop
Alternative flow	
Exception flow	[Database connection error] <ul style="list-style-type: none"> • Informs the user • Returns to main loop

Key components	Explanation
Name	Add Clause
Actors	User
Description	Add new clause into the KB
Goal	To add a new clause
Pre-condition	KB database exists
Trigger	User chose option Add Clause
Post-condition	New clause added to the image of KB in main memory
Normal flow	[User chose Add Clause] 1. The system asks for the clause 2. User enters clause 3. If the clause is in proper format, trigger Check Consistency 3.1. If successful, clause added, inform the user and return to main loop 3.2. Otherwise, clause dropped, inform the user and return to main loop
Alternative flow	[Invalid format] <ul style="list-style-type: none"> • Inform the user • Return to main loop
Exception flow	[Valid clause, but using undefined hedge definition] <ul style="list-style-type: none"> • Inform the user • Return to main loop [Database connection error] <ul style="list-style-type: none"> • Inform the user • Return to main loop

Key components	Explanation
Name	Edit Clause
Actors	User
Description	Edit a clause in the KB
Goal	To edit a clause
Pre-condition	KB exists
Trigger	User chose option Edit Clause
Post-condition	Clause edited
Normal flow	[User chose Edit Clause] <ol style="list-style-type: none"> 1. User enters clause's string 2. The system returns list of matched clause with id 3. User choose an id 4. User enter new clause 5. If the clause is in proper format, trigger Check Consistency <ol style="list-style-type: none"> 3.1. If successful, clause edited, inform the user and return to main loop 3.2. Otherwise, clause unchanged, inform the user and return to main loop
Alternative flow	[Invalid format] <ul style="list-style-type: none"> • Inform the user • Return to main loop [Unmatched string] <ul style="list-style-type: none"> • Inform the user • Return to main loop
Exception flow	[<i>Valid clause, but using undefined hedge definition</i>] <ul style="list-style-type: none"> • Inform the user • Return to main loop [<i>Database connection error</i>] <ul style="list-style-type: none"> • Inform the user • Return to main loop

Key components	Explanation
Name	Remove Clause
Actors	User
Description	Remove a clause in the KB
Goal	To remove a clause
Pre-condition	KB exists.
Trigger	User chose option Remove Clause
Post-condition	Clause removed from the KB
Normal flow	[User chose Remove Clause] 1. User enters clause's string 2. The system returns list of matched clause with id 3. User choose an id 4. Clause removed
Alternative flow	[No clause matched] <ul style="list-style-type: none"> • Inform the user • Return to main loop
Exception flow	[Null KB after removal] <ul style="list-style-type: none"> • Inform the user • Return to main loop [Database connection error] <ul style="list-style-type: none"> • Inform the user • Return to main loop

Key components	Explanation
Name	Check Consistency
Actors	User
Description	Check the consistency of a KB image
Goal	To check for a KB's consistency
Pre-condition	KB loaded
Trigger	User chose option Check Consistency
Post-condition	True returned if consistent, False if not.
Normal flow	[Check Consistency Triggered] 1. The KB is passed to the consistency checker 2. If the KB is consistent, return True 3. Otherwise return False
Alternative flow	
Exception flow	[Null KB] <ul style="list-style-type: none"> • Inform the user • Return to main loop
Key components	Explanation
Name	Query
Actors	User
Description	To ask the system to prove a clause
Goal	To see the confidence of truth value of the clause
Pre-condition	KB loaded
Trigger	User chose option Query
Post-condition	Return Nothing if the clause is false/unprovable Return Just <truth value> for the confidence of the clause's truth value
Normal flow	[Query Triggered] 1. User enters a clause. 2. If the clause is valid, perform resolution to prove it. 3. Return Nothing if the clause is false/unprovable, return Just <truth value> for the confidence of the clause's truth value 4. Ask the user if he wanted to print the explanation for the conclusion or not 4.1. If yes, trigger Explain Result 4.2. Otherwise, return to main loop
Alternative flow	[Invalid format] <ul style="list-style-type: none"> • Inform the user • Return to main loop
Exception flow	[Valid clause, but using undefined hedge definition] <ul style="list-style-type: none"> • Inform the user

- Return to main loop

[*Null KB*]

- Inform the user
- Return to main loop

Key components	Explanation
Name	Explain Result
Actors	User
Description	To explain a proof by printing the resolution trace
Goal	To see how the system prove the user's query
Pre-condition	KB loaded, Query triggered.
Trigger	User chose to Explain Result after Query printed the result
Post-condition	Return the trace of resolution process
Normal flow	[Explain Result Triggered] 1. The trace of the resolution process is printed
Alternative flow	
Exception flow	

Key components	Explanation
Name	Exit
Actors	User
Description	Terminate system
Goal	To terminate the system
Pre-condition	System executing
Trigger	User chose Exit
Post-condition	System terminated
Normal flow	[Exit Triggered] 1. System terminated
Alternative flow	
Exception flow	