On Reformulating Planning As Dynamic Constraint Satisfaction

Jeremy Frank
Ari Jonsson
Paul Morris

NASA Ames Research Center

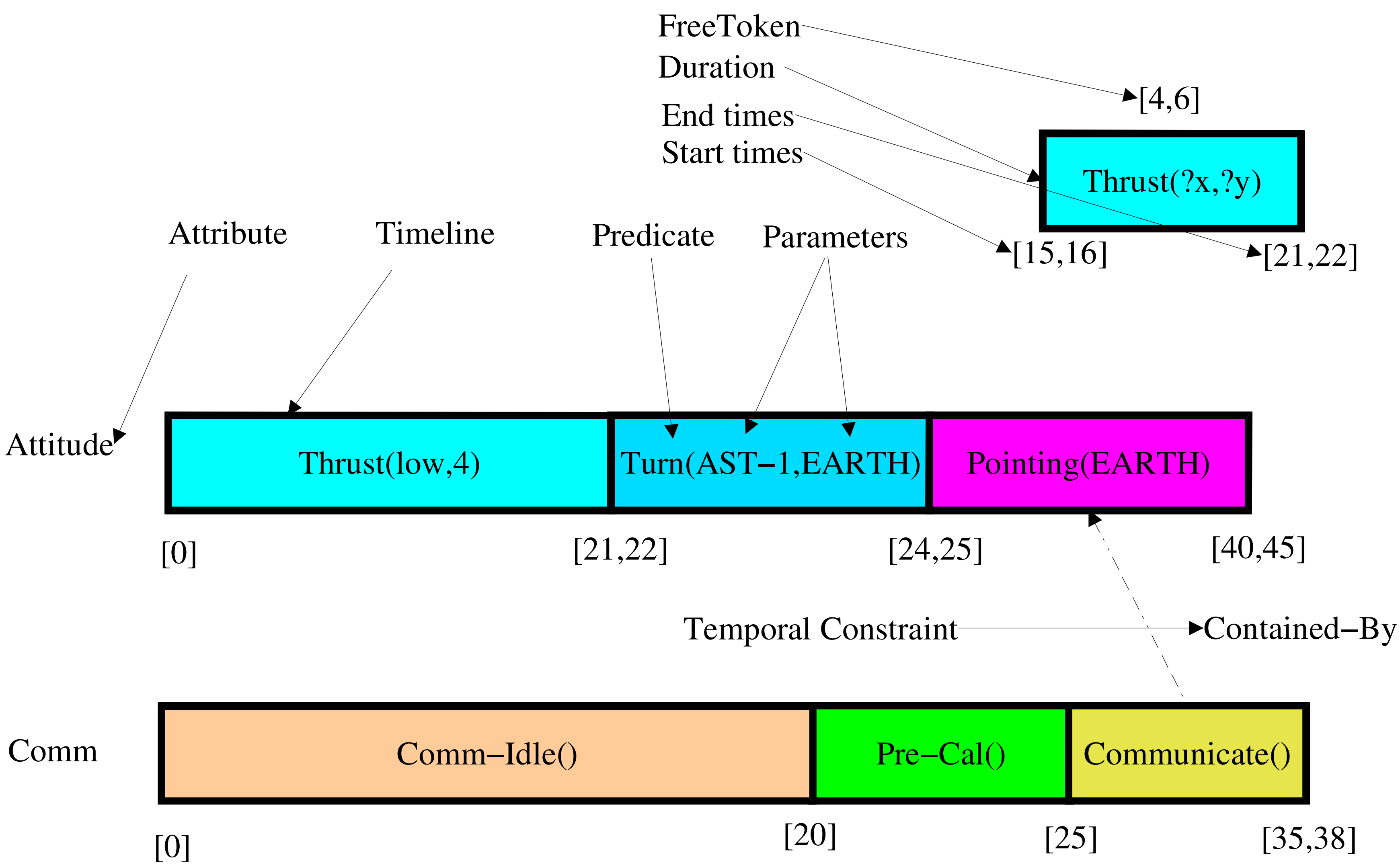
Today's Presentation

- Constraint–Based Interval Planning
  - Reformulating planning as constraint satisfaction
- Attributes and Mutual Exclusion Enforcement
- Two Formulations
  - Slot–based
  - Token ordering
- Comparing Formulations
- Discussion

CBIP: Domain Models

- Attributes – aspect of the world or system
  - Timeline – history of values of the attribute
- Token – value of an attribute over an interval
  - Predicate and Parameters
  - Uniform represention of actions and fluents
  - Free tokens
- Planning Axioms
  - Specify constraints on parameters of predicates
  - Specify relations between tokens

CBIP: Domain Models



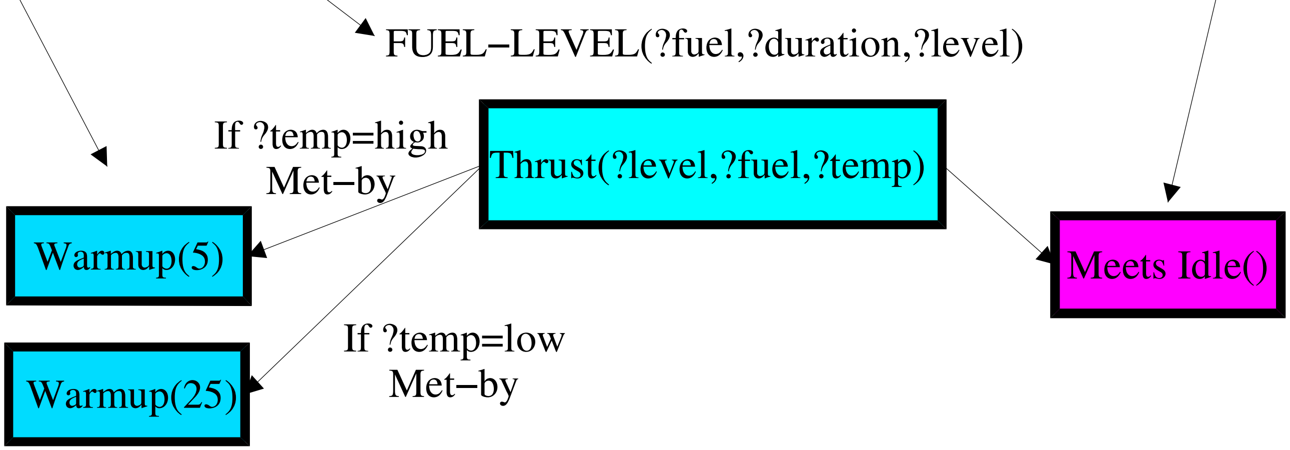
Example: A Simple Plan Axiom

```
Thrust(?level,?fuel,?temp) :-
    FUEL_LEVEL(?fuel,?duration,?level)
    Meets Idle()
    OR ?temp
        high: Met-By Warmup(5)
        low: Met-by Warmup(25)
```

Parameter Constraints

Interval Constraints

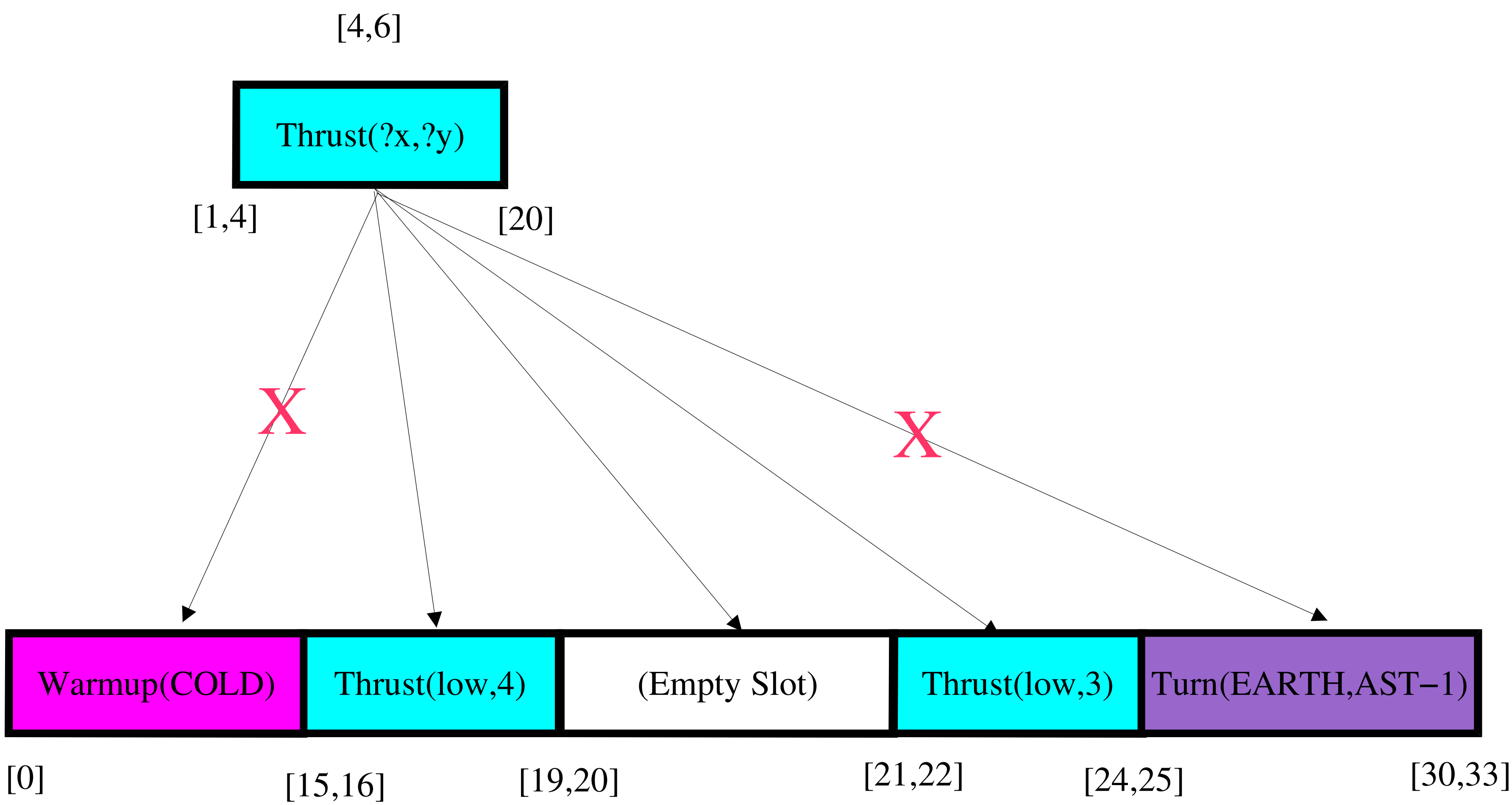
Disjunctive Constraints



CBIP: Attributes and Mutual Exclusion

- Inserting tokens onto a timeline extends plan
- Timelines enforce mutual exclusion
  - Tokens with different predicates must be totally ordered
  - Tokens with identical predicates may be equated
- Slot – legal location for a token
  - Tokens can be inserted onto a single slot
  - Full slot – token already occupies this space
  - Empty slot – no token occupying this space

Example: Adding a Free Token



Underlying Representation: DCSPs

- Constraint Satisfaction Problem (CSP)
  - Variables, Domains, Constraints on assignments
- Dynamic CSPs (DCSPs)
  - Restriction – more variables, fewer values, tighter constraints
  - Relaxation – fewer variables, more values, looser constraints
- Solving CSP means satisfying all constraints
- CSP is consistent if it has a solution

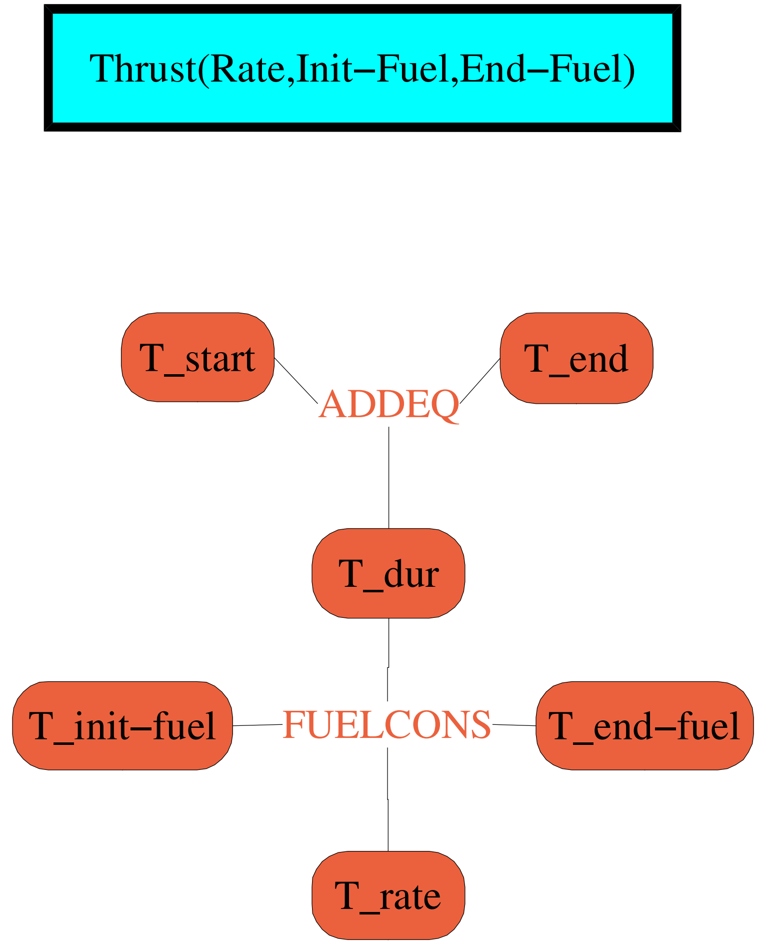
Representing a Plan as a DCSP

- Each token represented by a set of variables
  - Temporal variables: start, end, duration
  - Parameter variables (e.g. Rate of thrust)
- Plan axioms define constraints between variables
  - Simple temporal relations lead to temporal constraints
  - Equivalence between parameters of different tokens
  - Arbitrary constraints on parameters
    - Example: fuel consumption constrains rate, duration, fuel

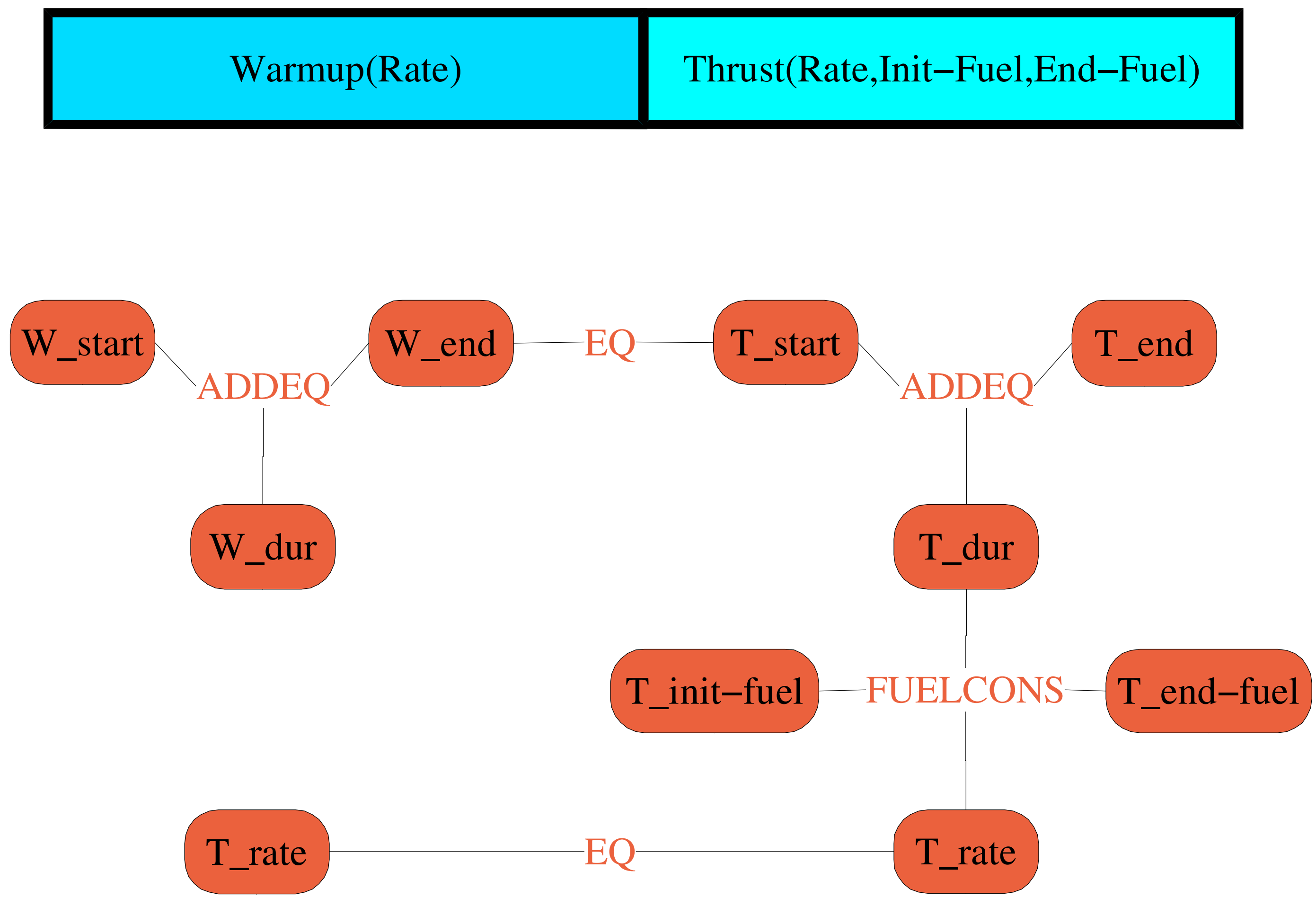
Representing a Plan as a DCSP (cont)

- As planning proceeds, candidate plan changed
  - Tokens added/deleted from attribute timelines
  - Constraints added/removed
- Results in a DCSP
  - Variables added/removed
  - Constraints added/removed
- Rather than use plan axioms directly, use CSP techniques
  - If CSP is inconsistent, the current plan candidate is invalid

Example: Adding Action to the Plan...



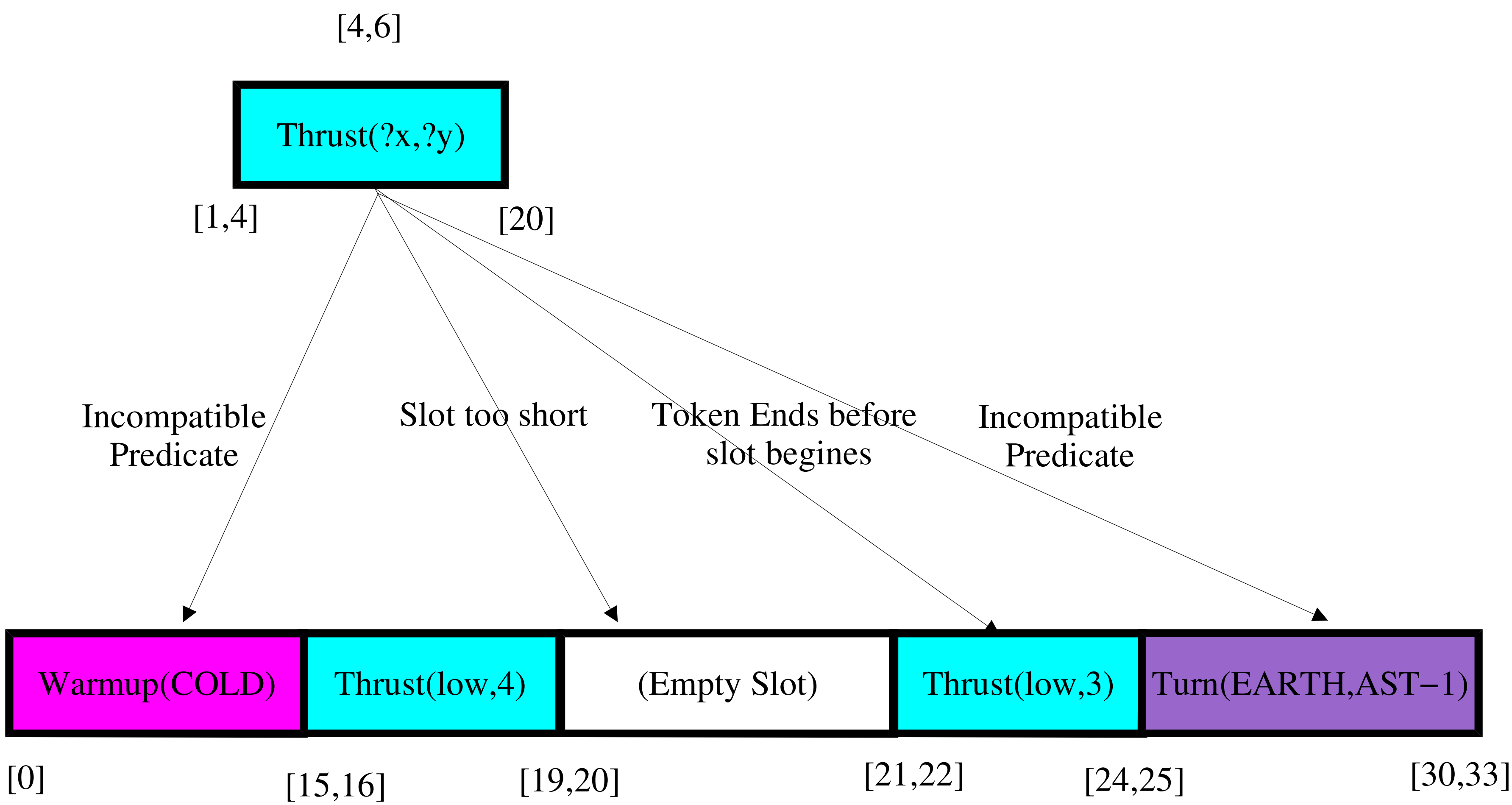
...Adds Variables and Constraints



Representing Mutual Exclusion I

- Must reason about where to insert free tokens
- Slot–based representation:
  - Variable representing available slots
  - One value per slot
  - Use data structure to represent slots on each timeline
- Not all slots are possible
  - Clashing predicates
  - Temporal constraints or parameter constraints
  - Use special purpose lookahead to eliminate slots

Example:Lookahead to Eliminate Slots



Representing MutEx I (cont)

- As planning proceeds, possible slots may change
  - Adding tokens to timelines creates and destroys slots
  - New temporal constraints may affect candidate slots
  - Slots are a dynamic domain with changing value labels

Example:A Problem?

Thrust(?x,?y) Idle(5)

Not(Thrust, Empty Slot X)

Warmup(COLD)	Thrust(low,4)	(Empty Slot X)	Thrust(low,3)	Turn(EARTH,AST−1)

Thrust(?x,?y)

Not(Thrust, ???)

Warmup(COLD)	Thrust(low,4)	(Empty Slot Y)	Idle()	(Empty Slot Z)	Thrust(low,3)	Turn(EARTH,AST−1)

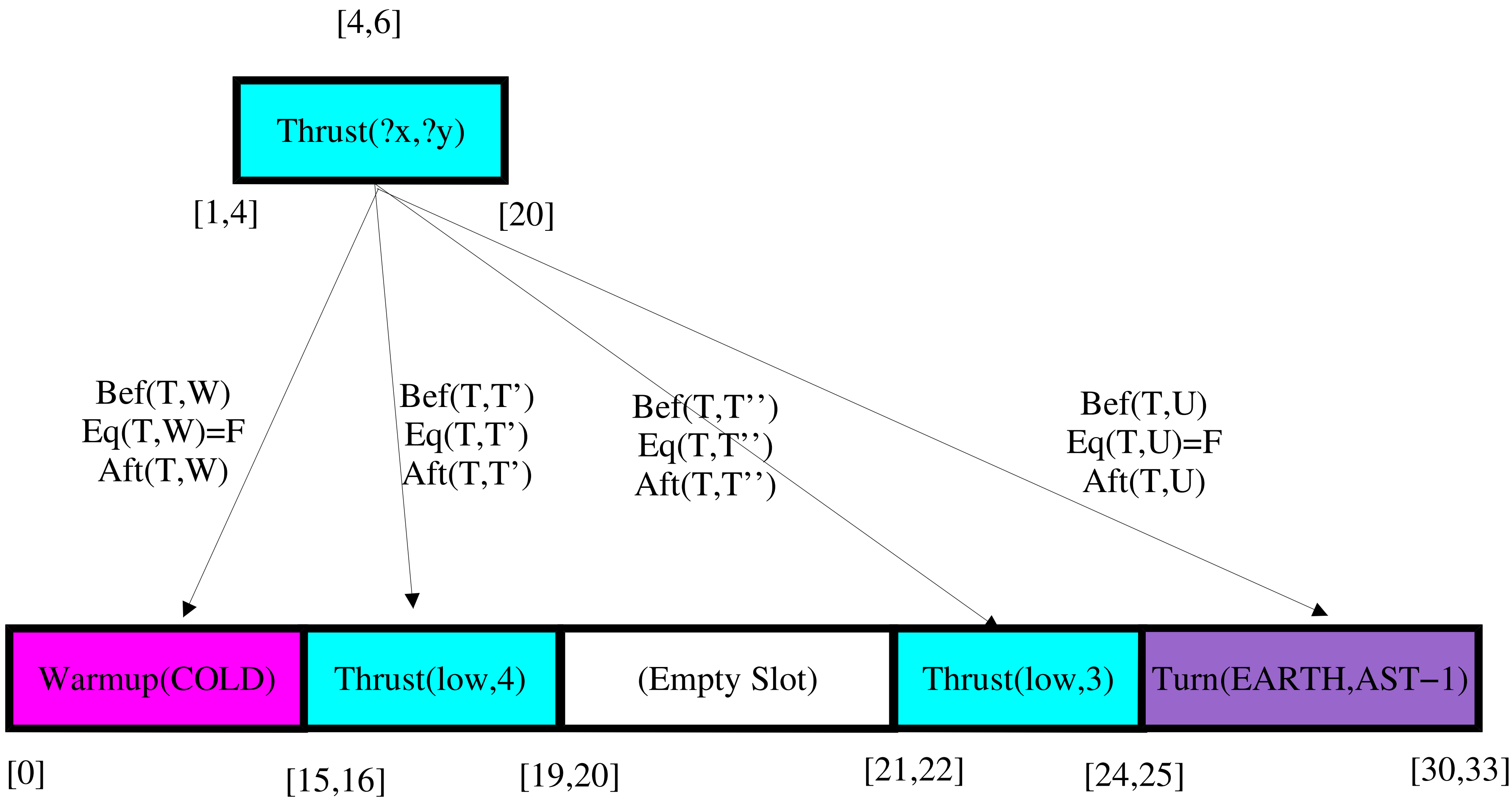
Representing Mutual Exclusion II

- Timelines are ordered lists of token equivalence classes
- Slots are consequence of committing to token ordering
- Use variables to represent ordering decisions
  - Logical variables *Bef(A,B), Aft(A,B) Eq(A,B)* represent possible ordering decisions
  - Domains of ordering variables are not dynamic
  - Nogood issue for token insertions vanishes

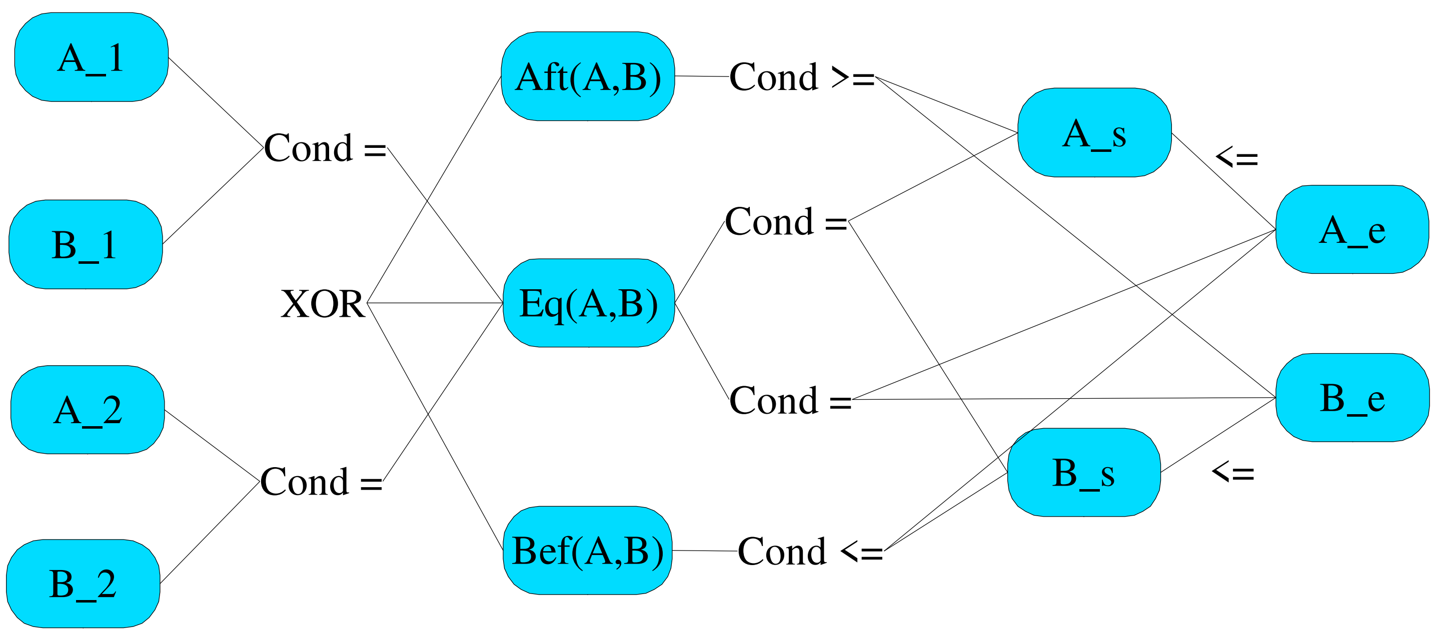
Representing MutEx II (cont)

- Conditional constraints
  - Link ordering variables with variables representing tokens
  - Enforce following conditions:
    - $Bef(A,B) \Leftrightarrow A_e \leqslant B_s$
    - $Aft(A,B) \Leftrightarrow B_e \leqslant A_s$
    - $Eq(A,B) \rightarrow (A_e = B_e) \wedge (A_s = B_s) \wedge (a_i = b_i)$
  - Ensures propagation of information from temporal variables (token start, end) to logical variables
- Since only one ordering can hold, post XOR constraint
- If A, B have incompatible predicates, post $\neg Eq(A,B)$

Example:Ordering Variables



Example: The Induced DCSP



Features of Representation

- Lookahead subsumed by arc−consistency
  - Incompatible predicates handled by unary constraint
  - Temporal constraints handled by the conditional constraints reducing domains of the logical variables
  - Conditional constraints on equating tokens