

How to Remember and Revisit Many Genetic Design Variants Automatically



Nicholas Roehner, Douglas Densmore
CIDAR Lab of Boston University

COMBINE 2016 | Sept. 19 - 23 | Newcastle Upon Tyne

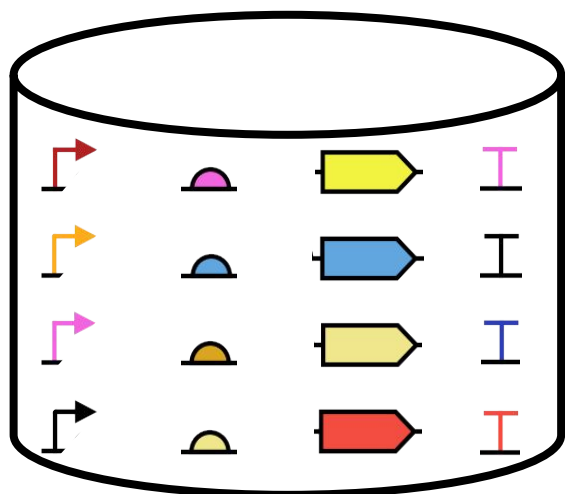


Splice: Synthetic Biology Board Game

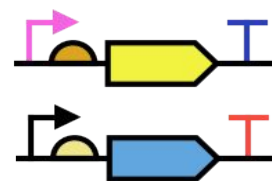




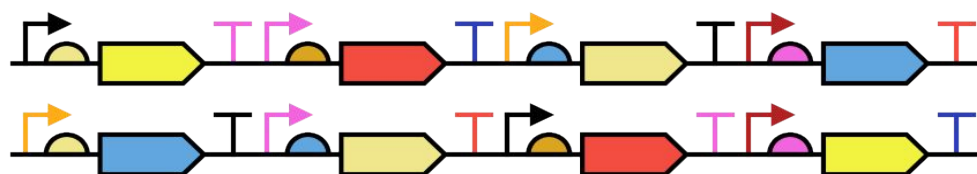
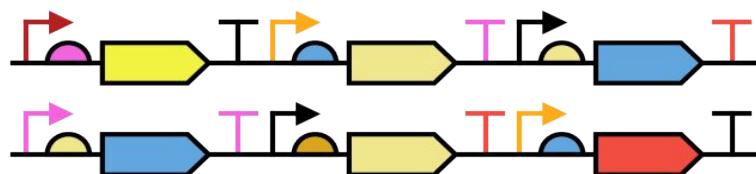
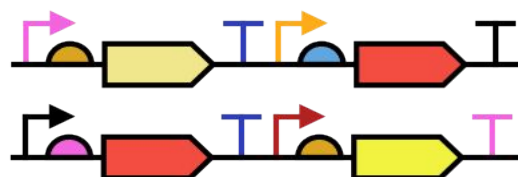
How to Encode Many Possible Designs for a 4-Gene Cluster?



16 Parts

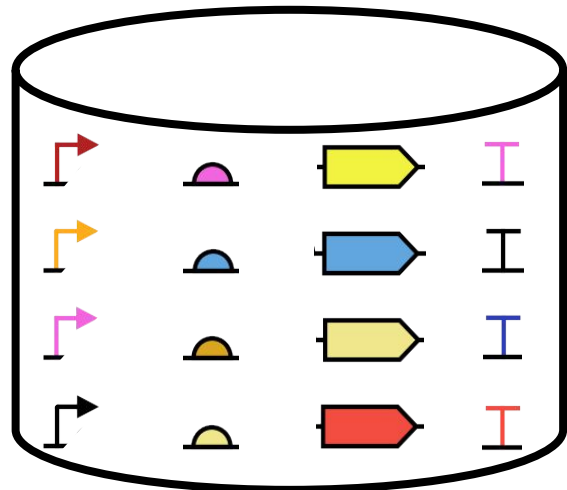


684,544
Design Variants

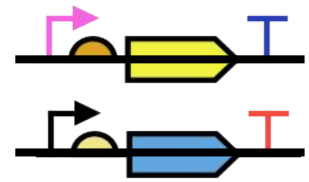




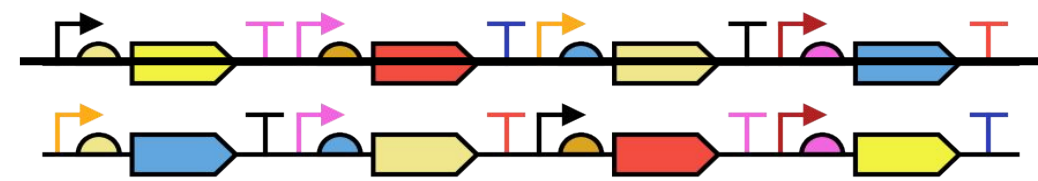
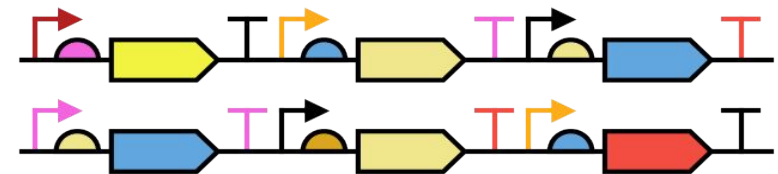
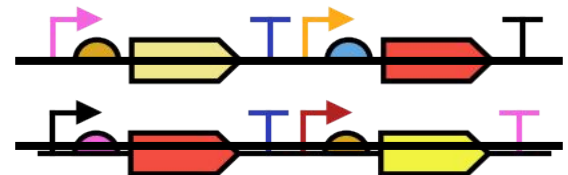
How to Track Which Designs are Ruled Out as Data is Gathered?



16 Parts

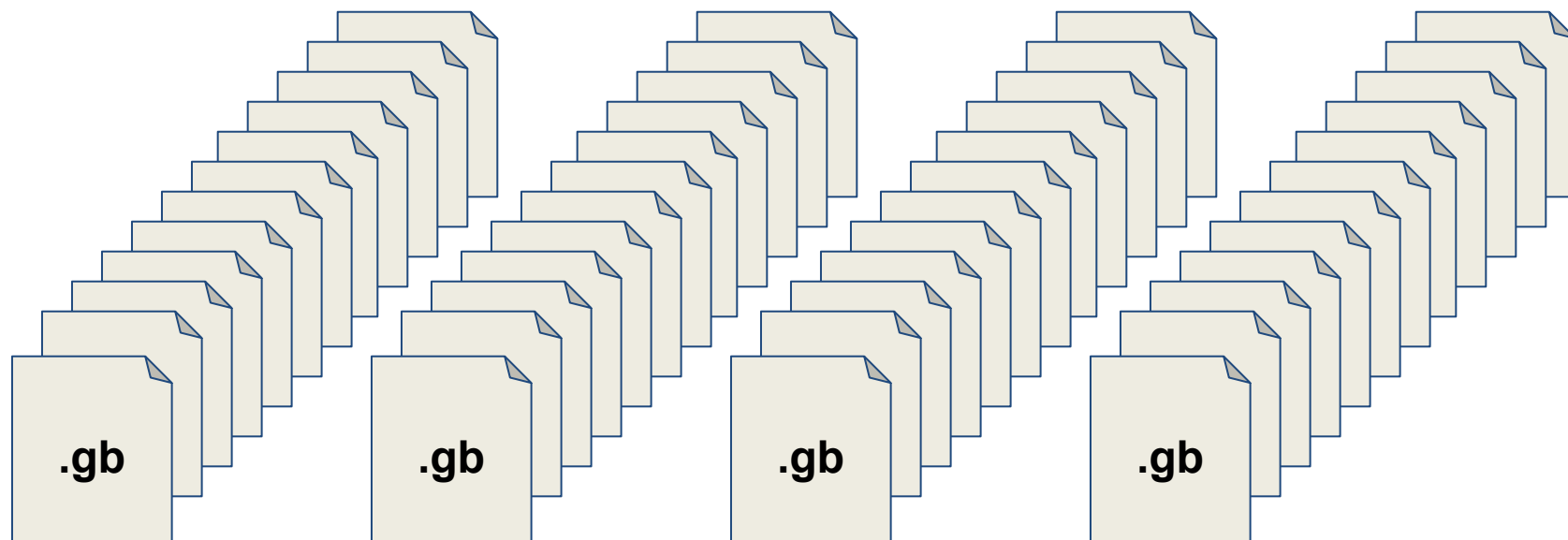


684,544
Design Variants





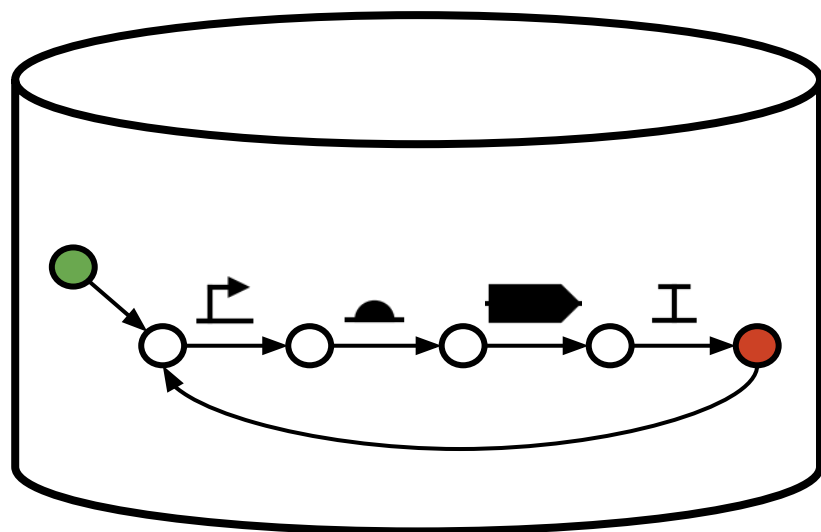
GenBank is Not the Answer



684,544 Files
Memory >6 GB

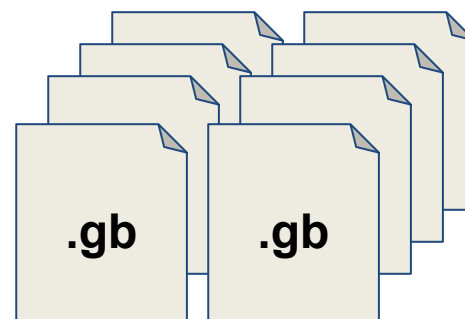


Knox Provides a Solution



Genetic Design Space
Memory <200 KB

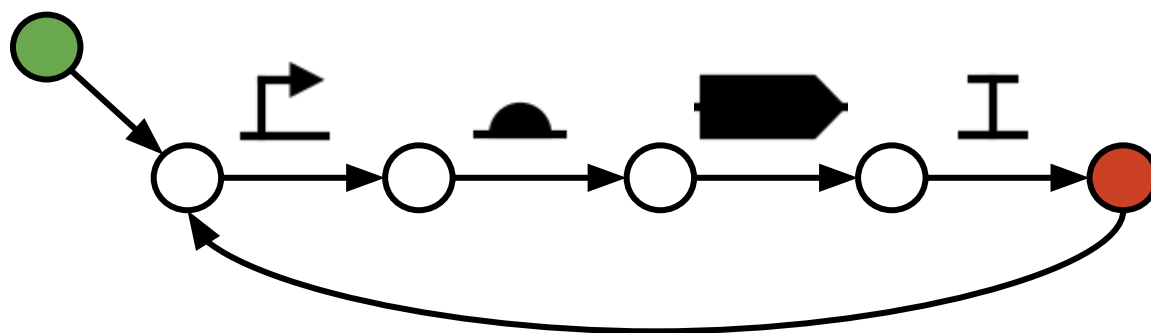
=



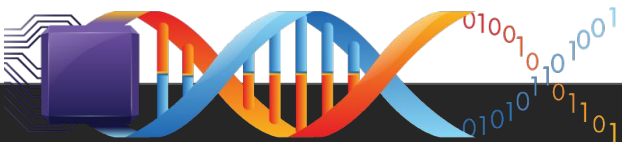
684,544 GenBank Files
Memory >6 GB



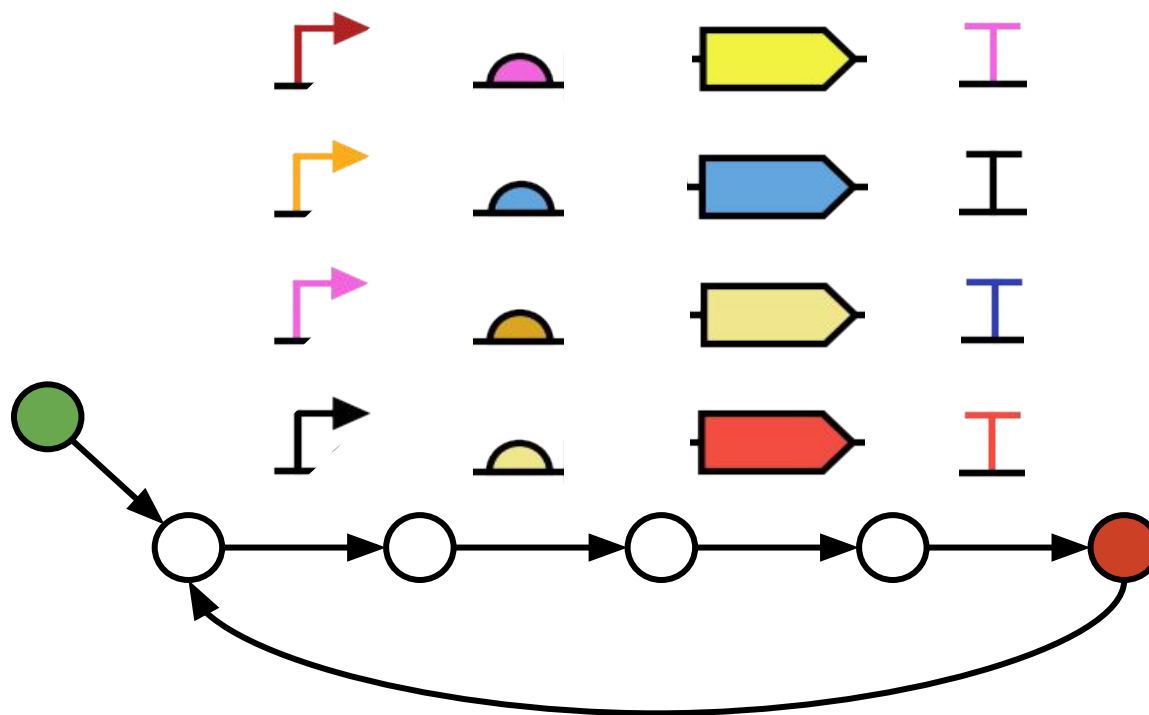
What is a Genetic Design Space?



A genetic design space is a graph that encodes design rules for composing sets of genetic parts.

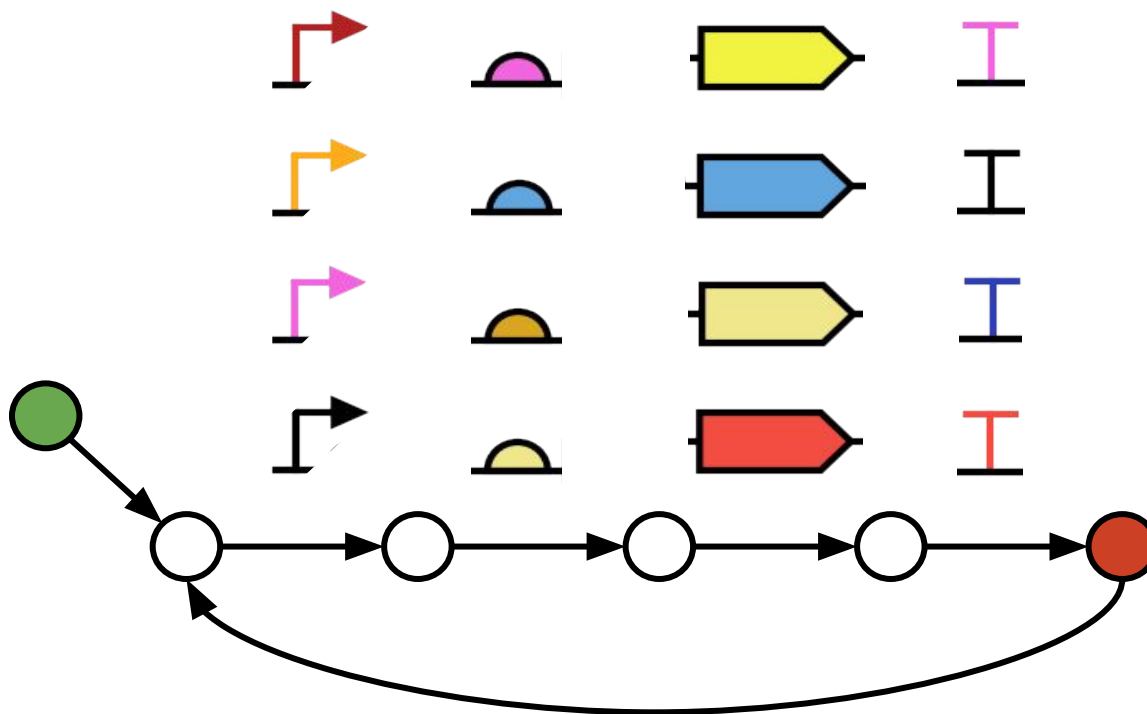


Each Edge is Labeled with a Set of Parts



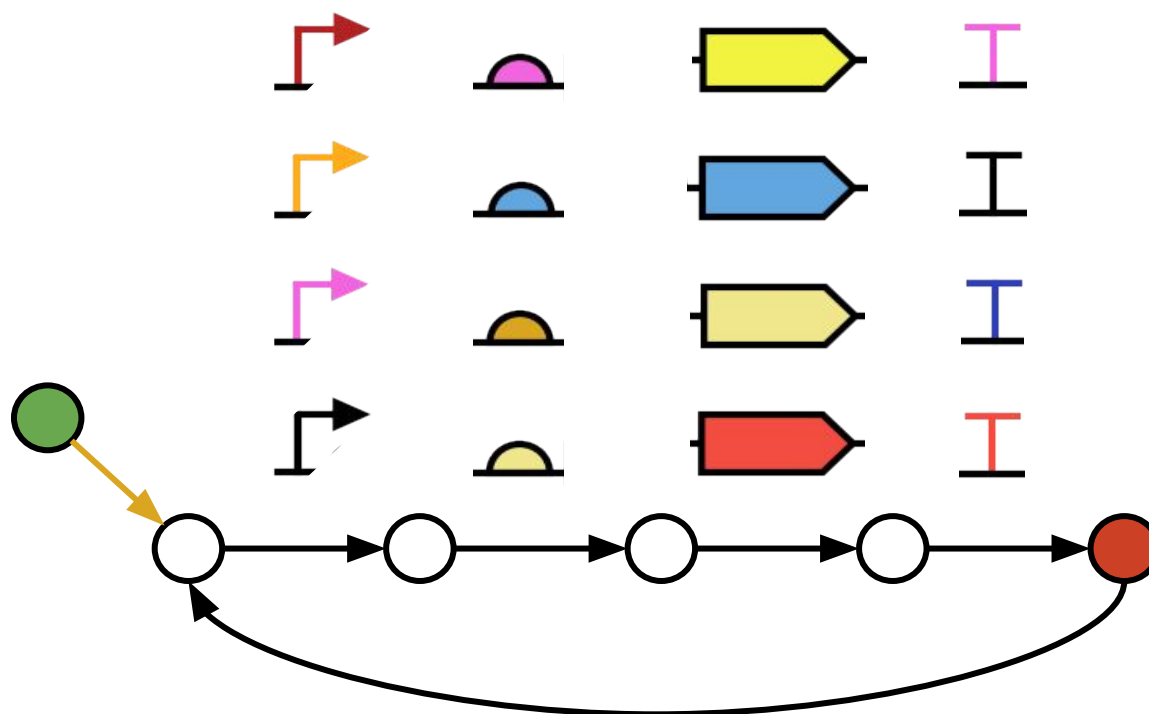


Path for Single Gene



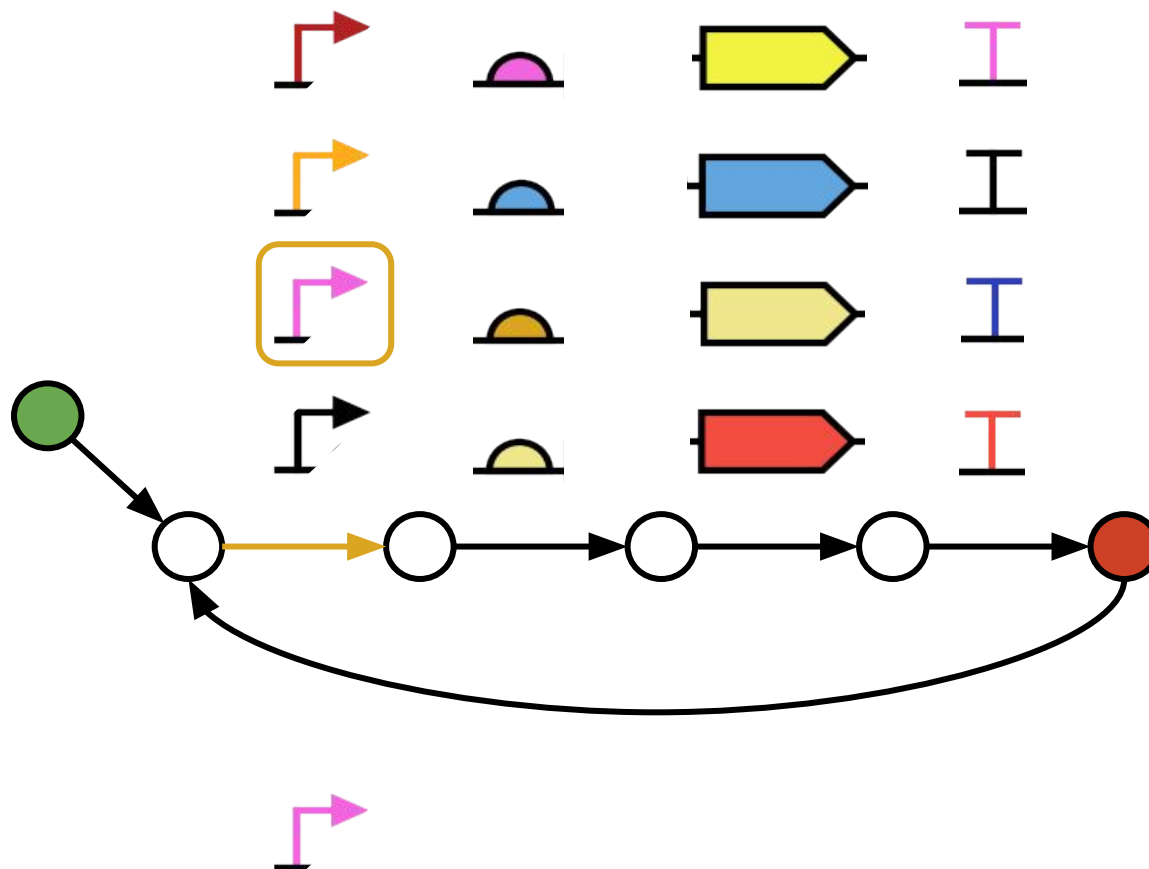


Path for Single Gene



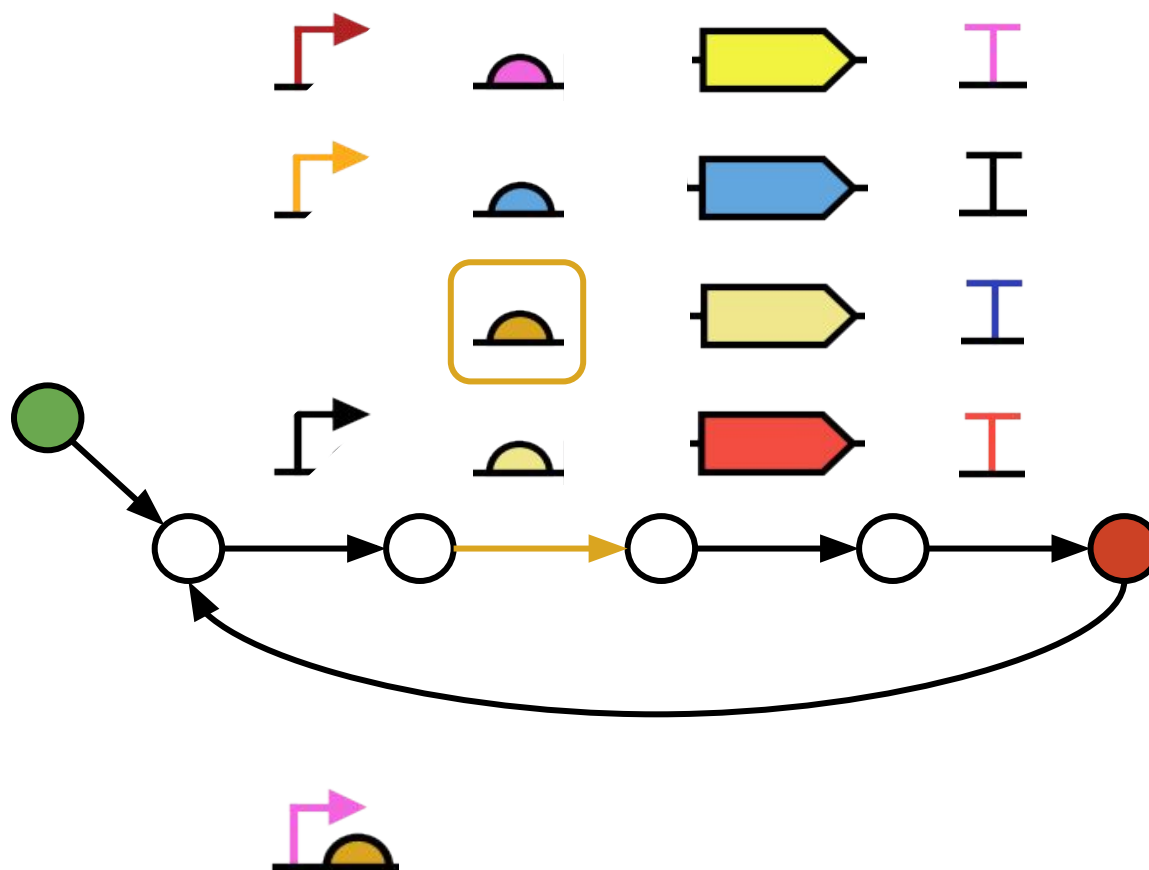


Path for Single Gene



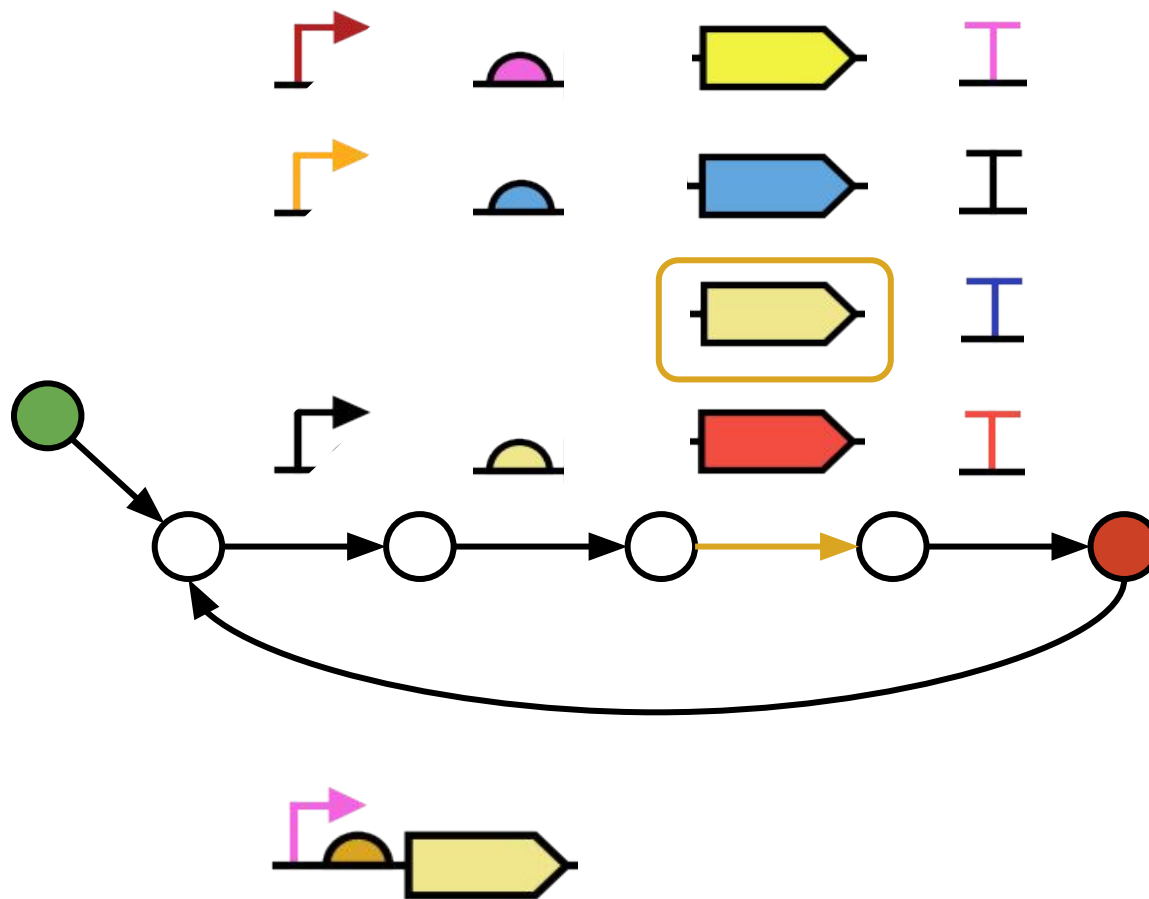


Path for Single Gene



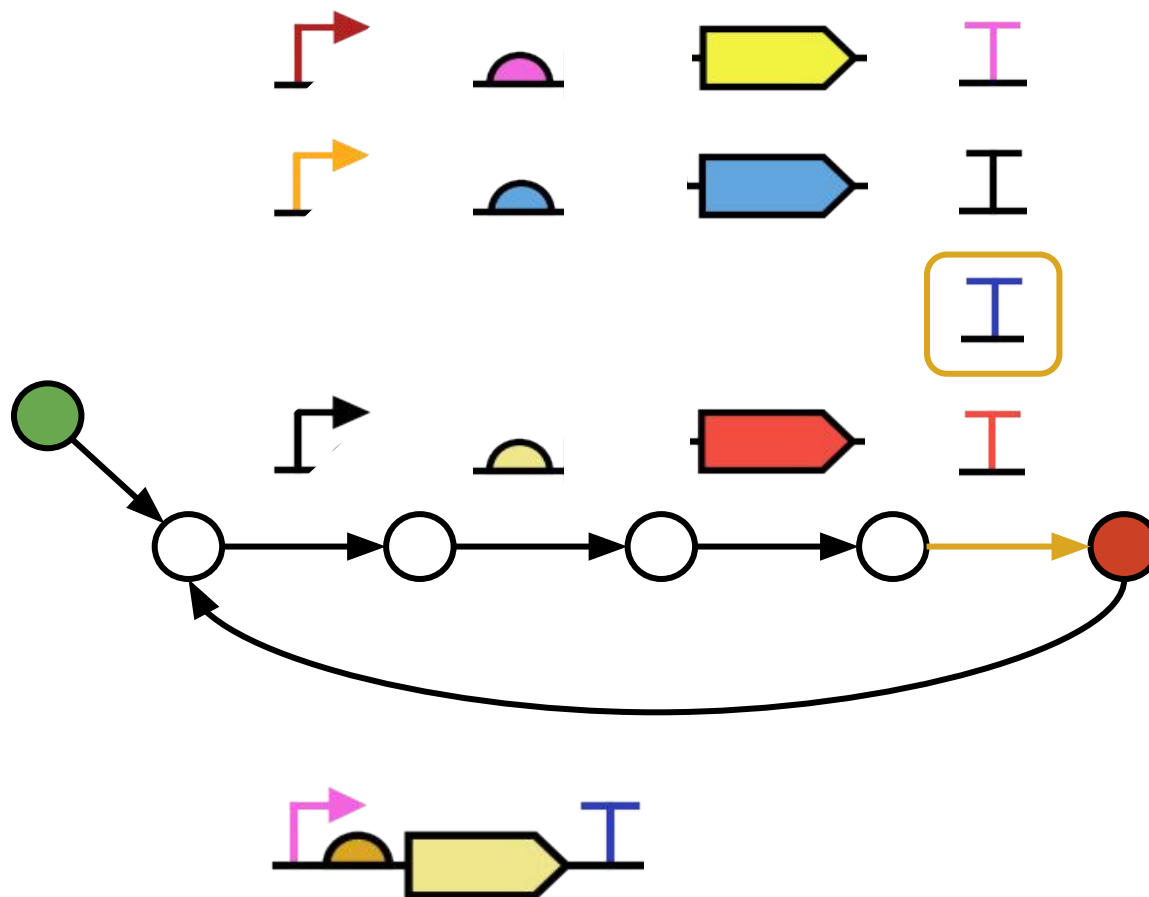


Path for Single Gene



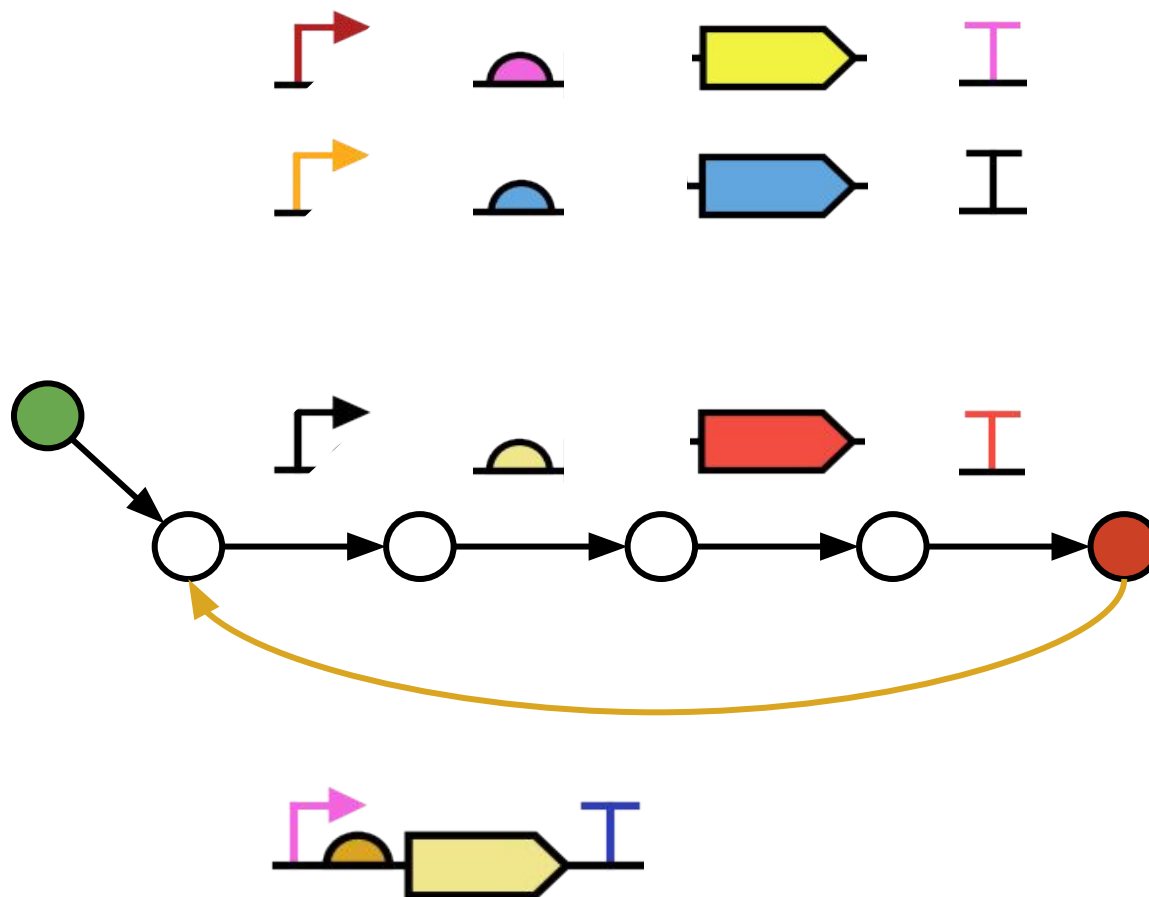


Path for Single Gene



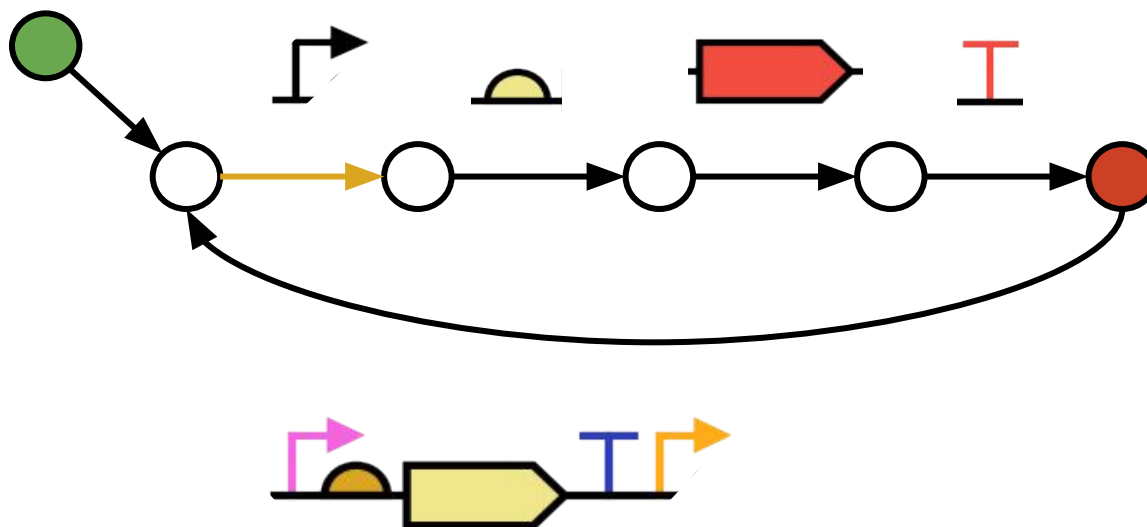
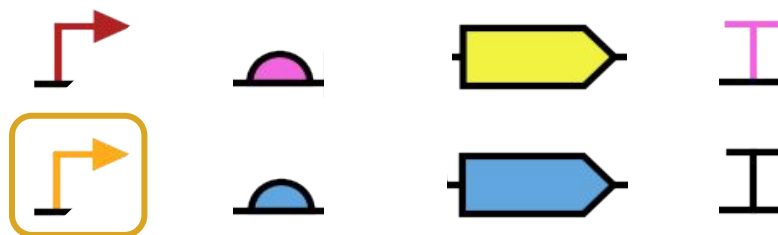


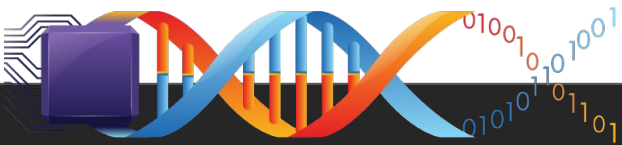
Path for Two Genes



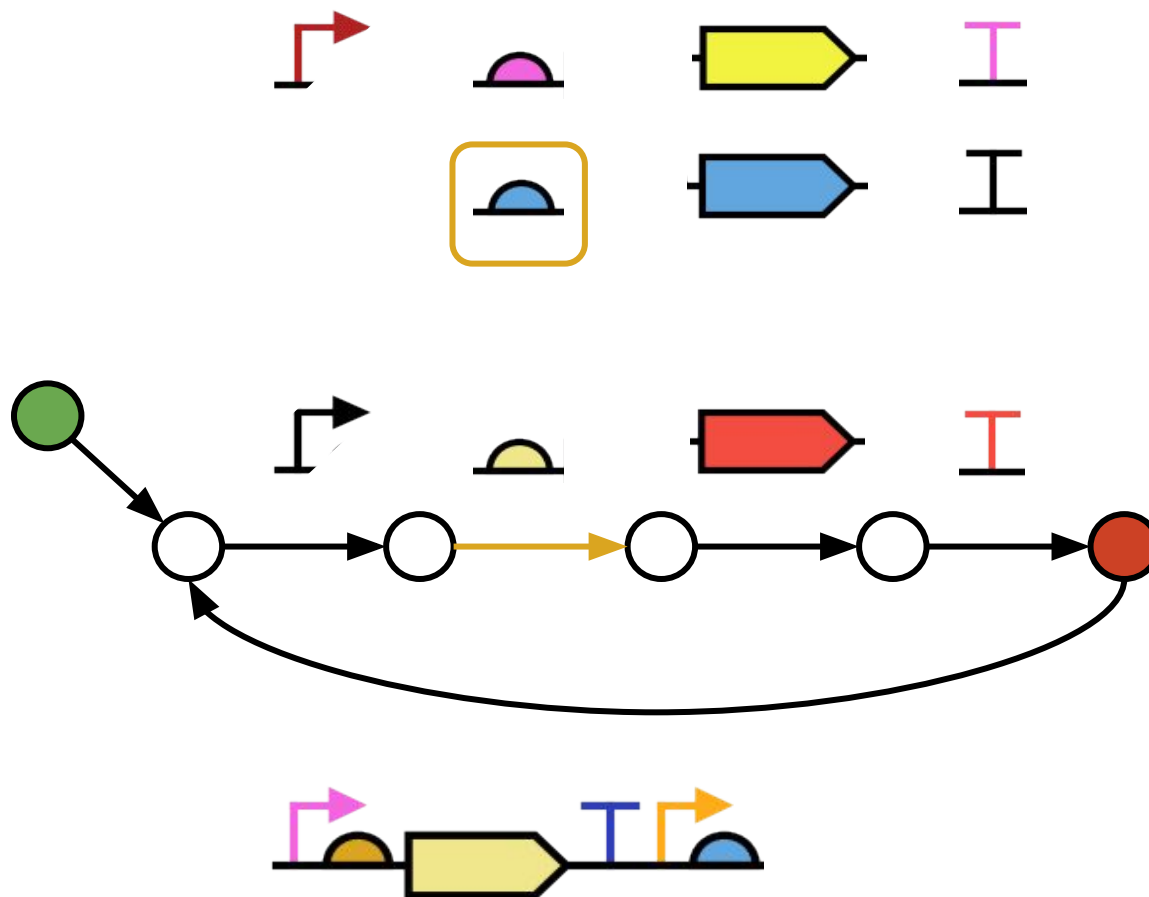


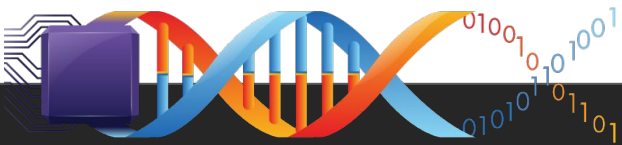
Path for Two Genes



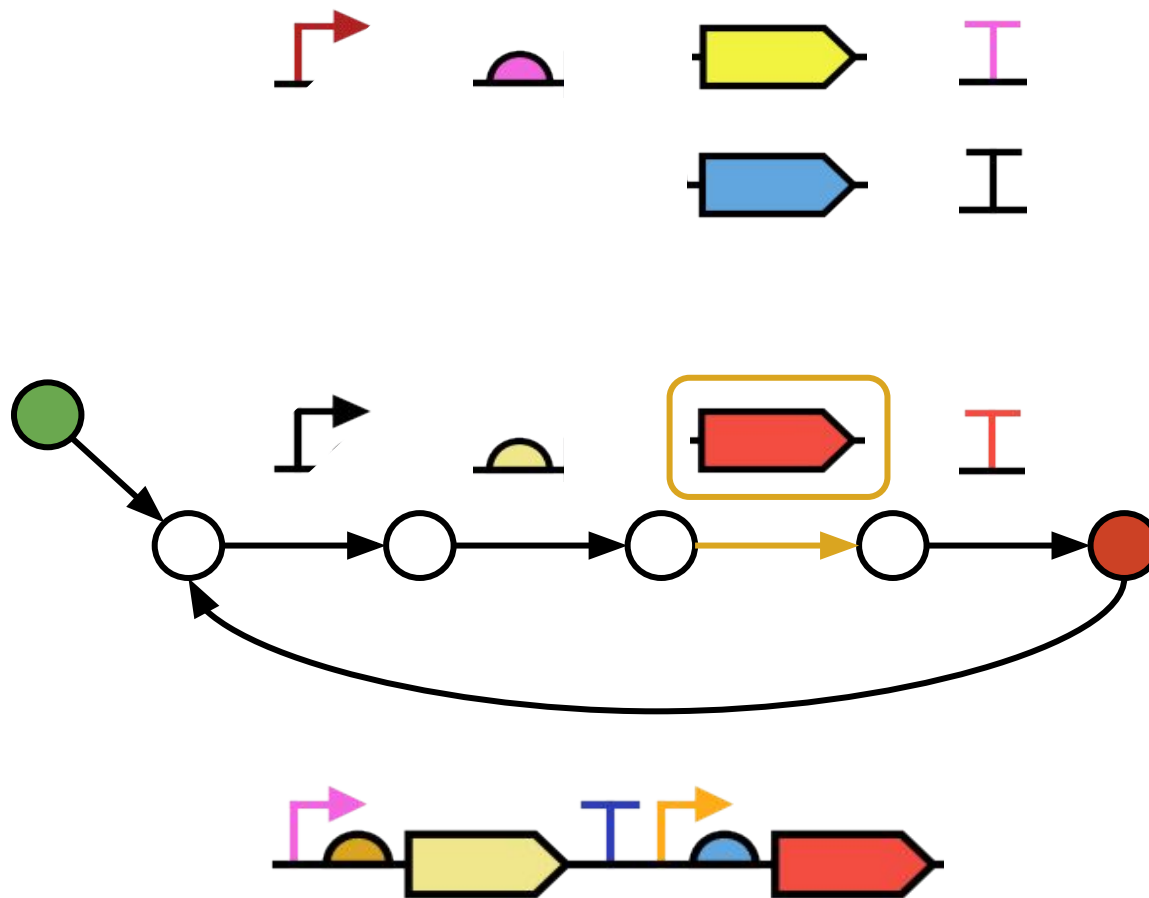


Path for Two Genes



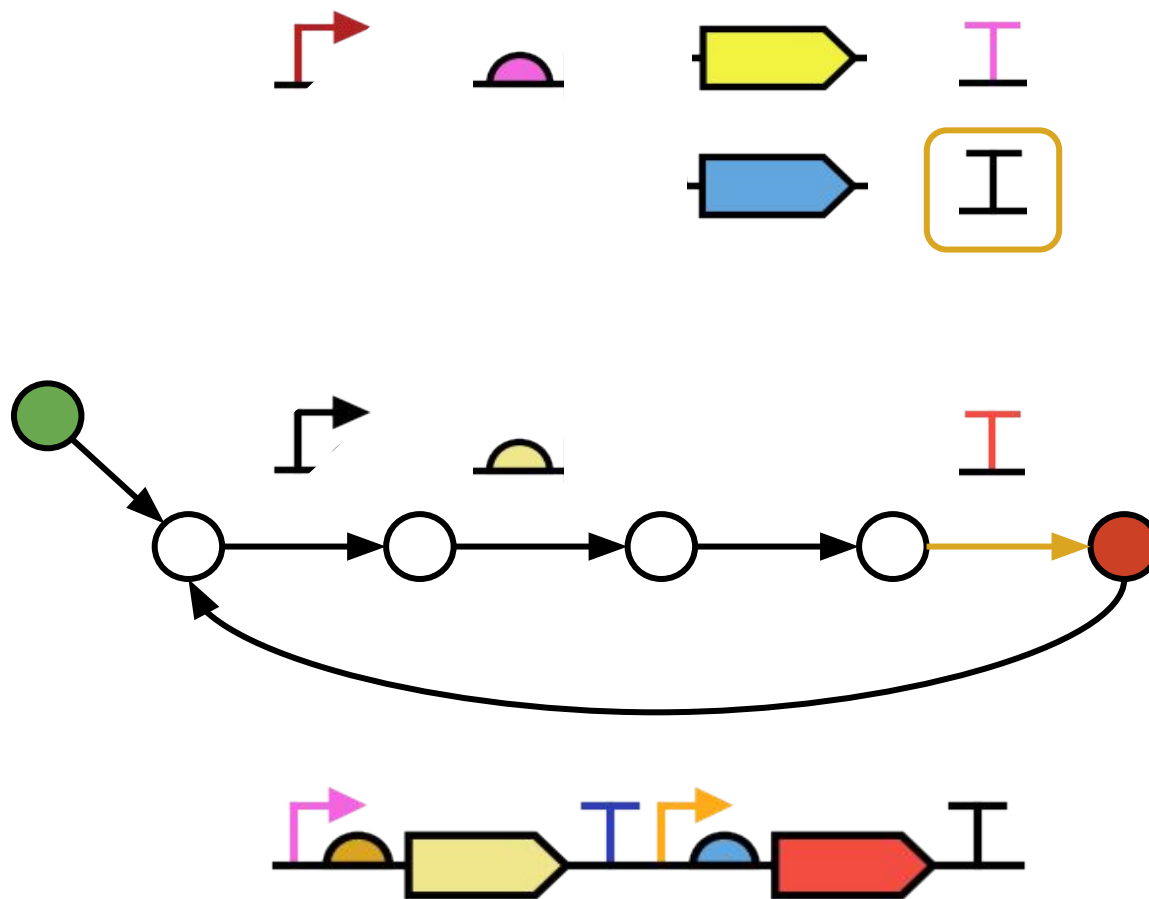


Path for Two Genes



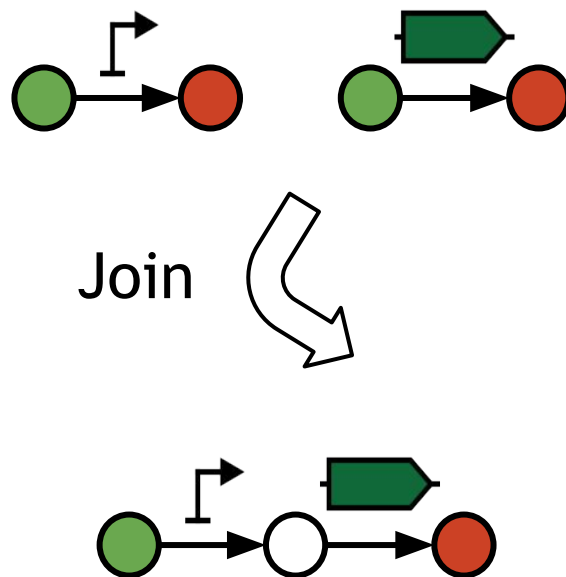


Path for Two Genes



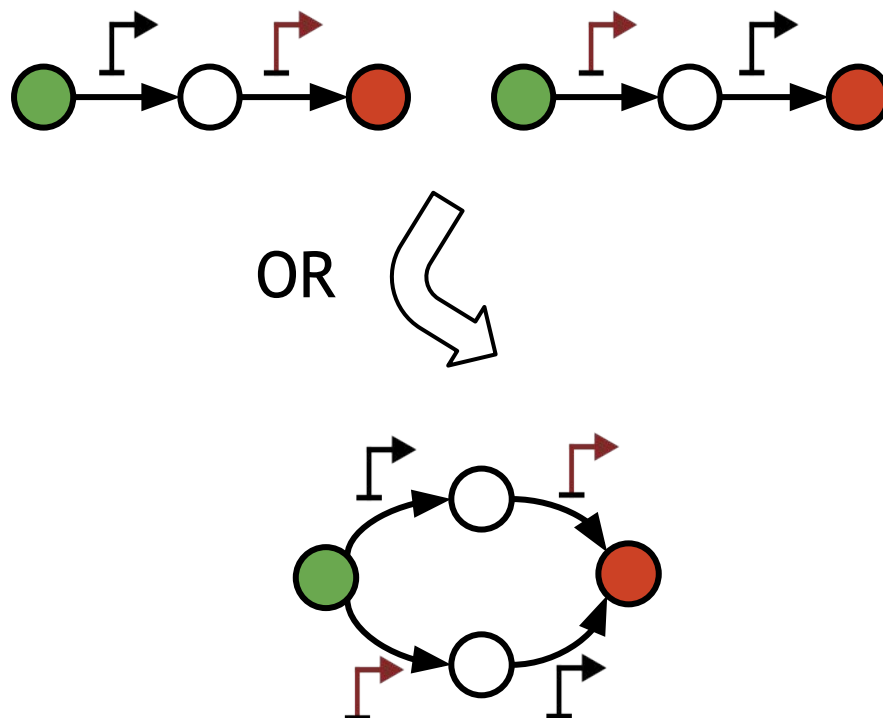


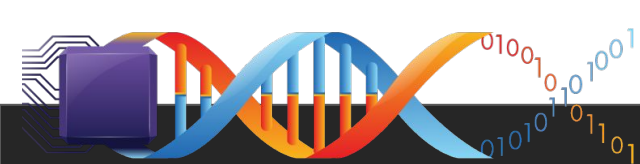
Design Spaces Can Be Combined via Graph Operations in Knox



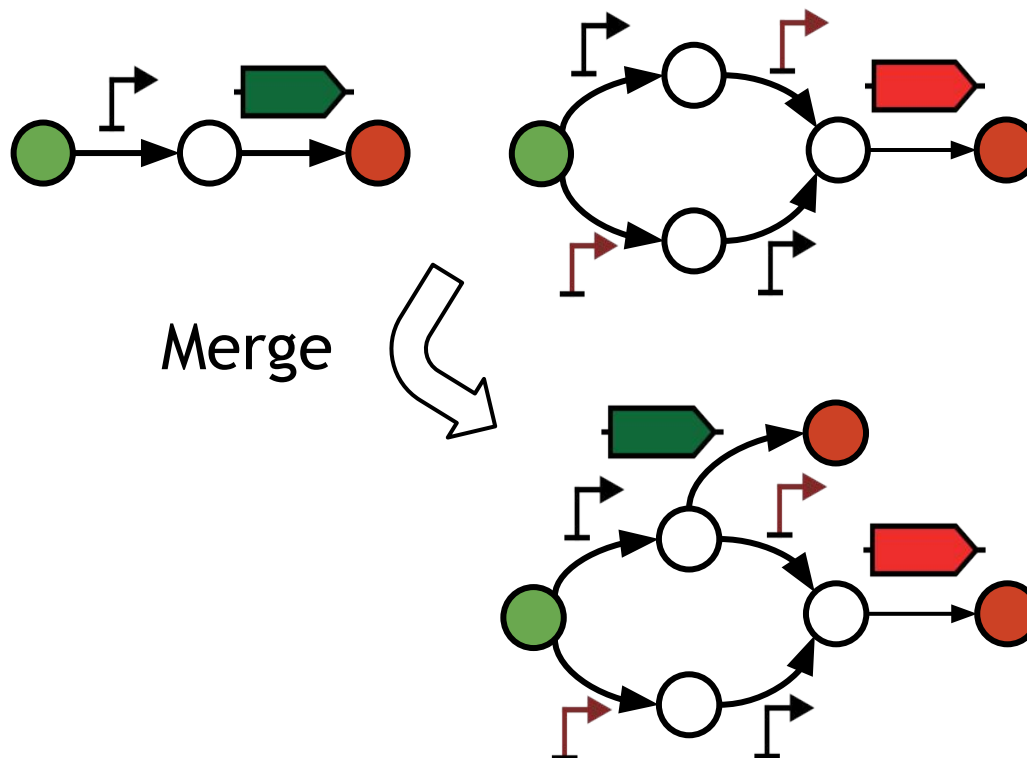


Design Spaces Can Be Combined via Graph Operations in Knox



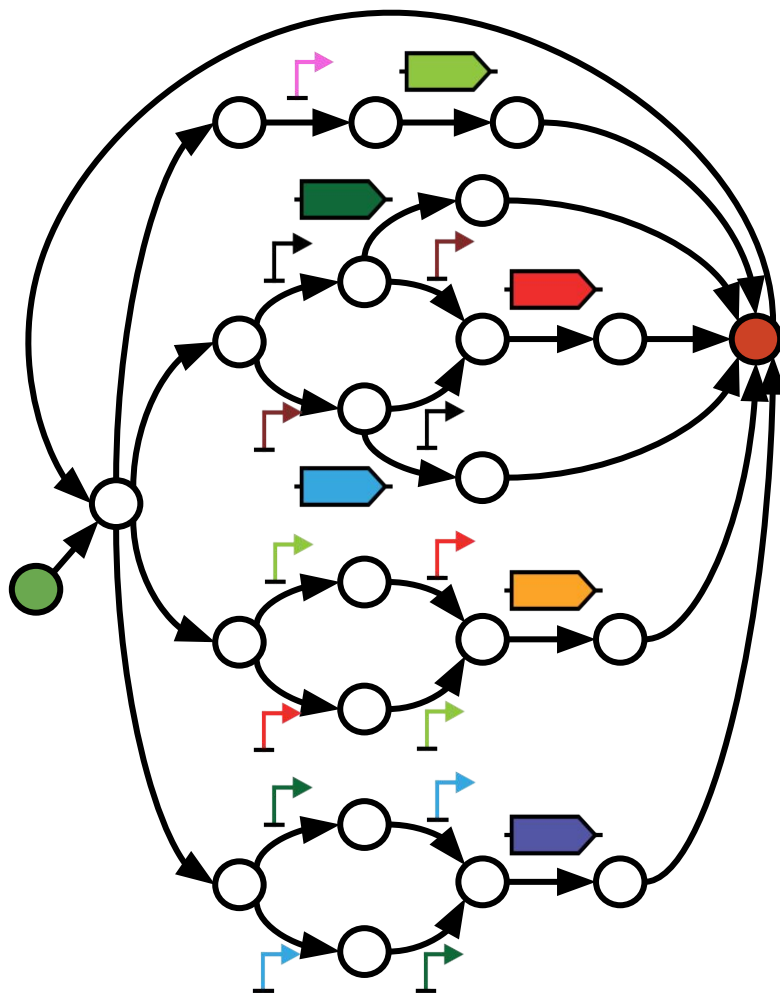


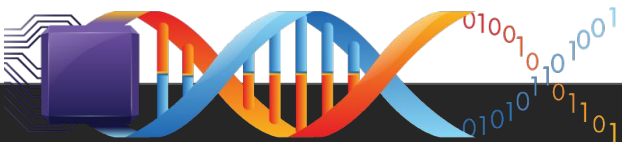
Design Spaces Can Be Combined via Graph Operations in Knox



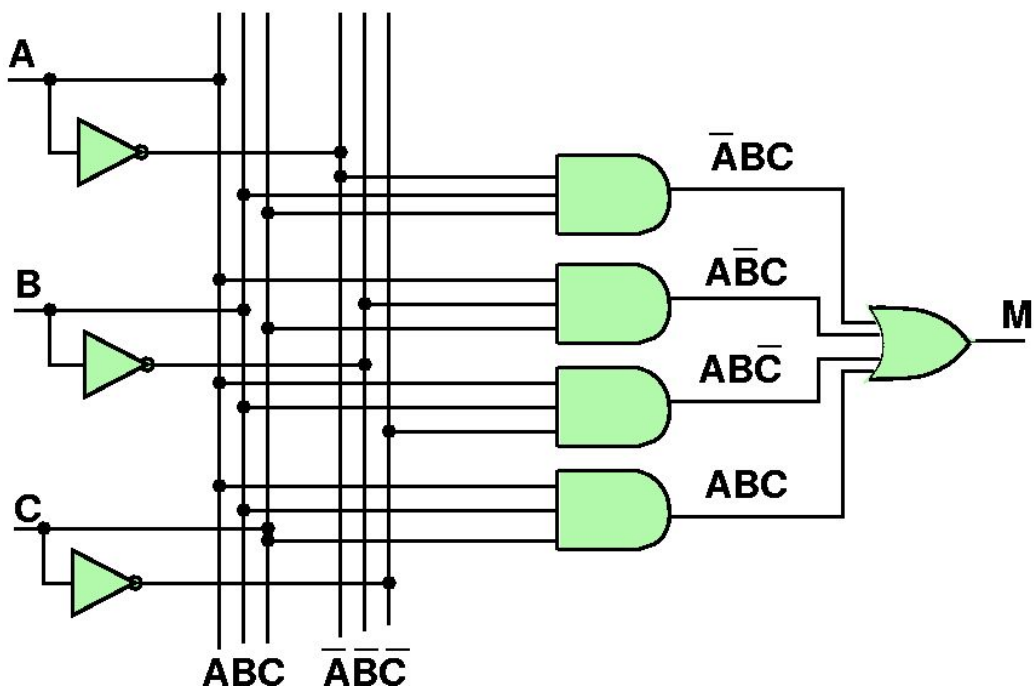
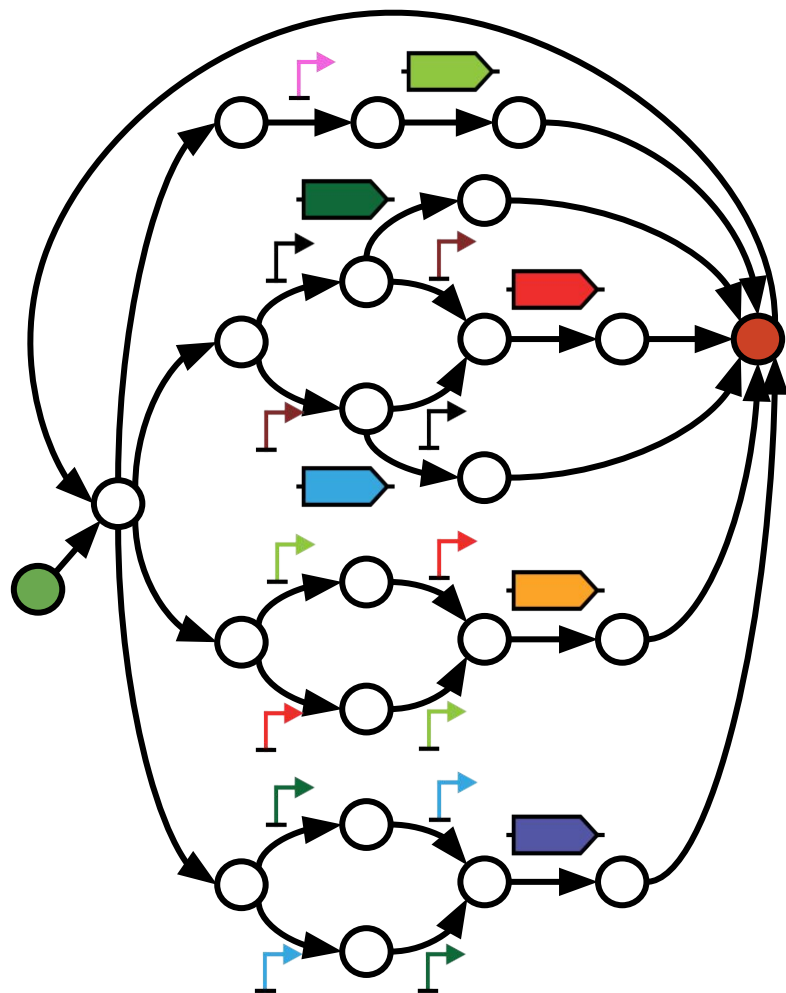


Design Space for Majority Circuit



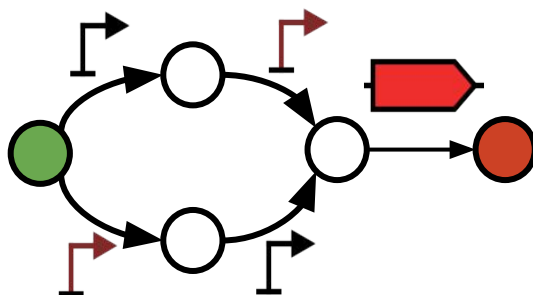
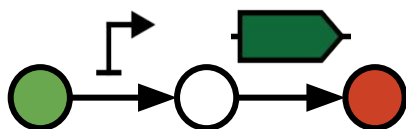


Design Space for Majority Circuit

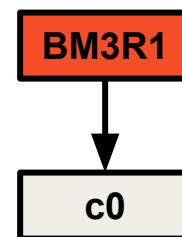
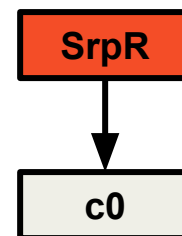




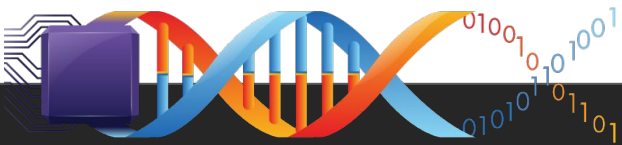
Every Design Space in Knox has a Version History



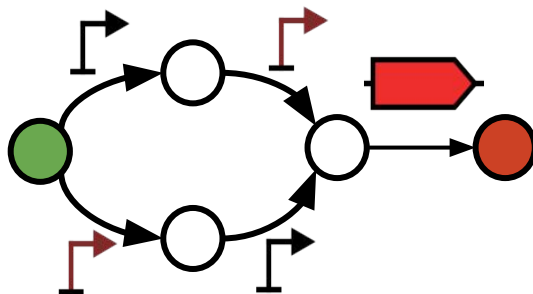
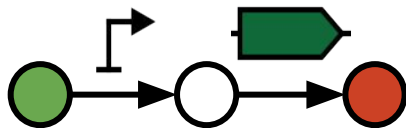
Design Spaces



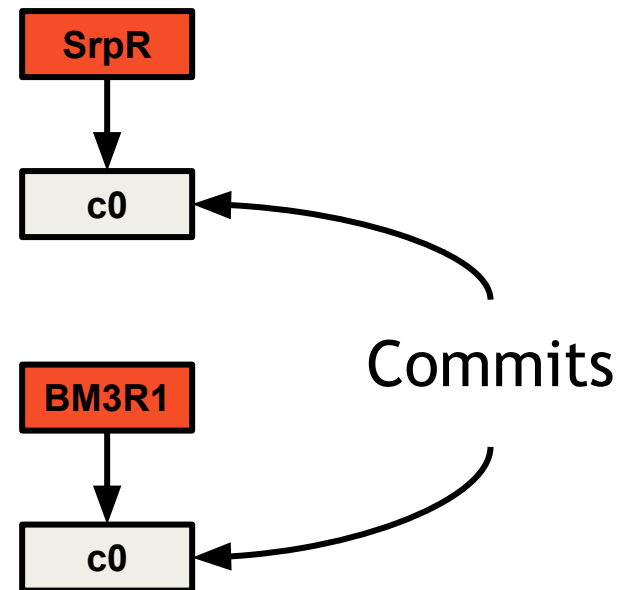
Version Histories



Every Design Space in Knox has a Version History



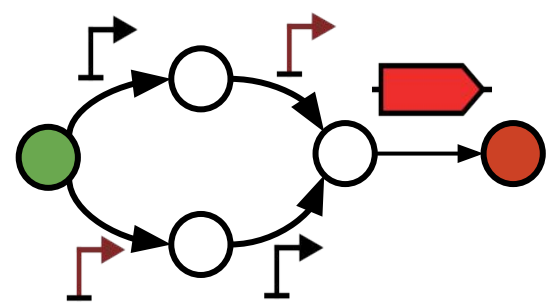
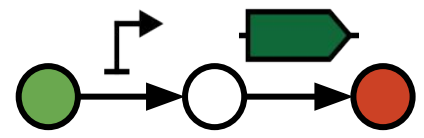
Design Spaces



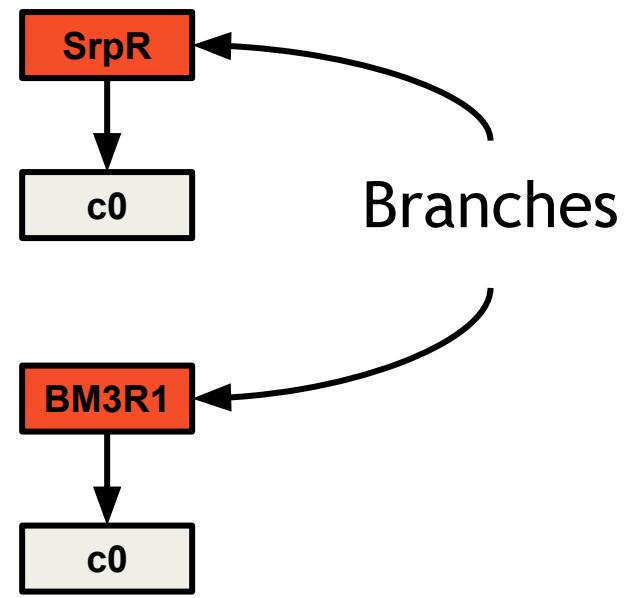
Version Histories



Every Design Space in Knox has a Version History



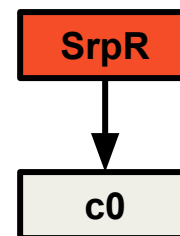
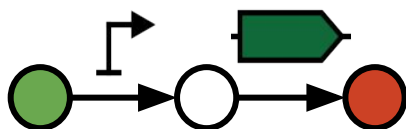
Design Spaces



Version Histories

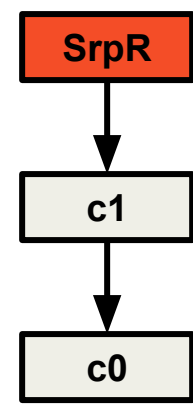
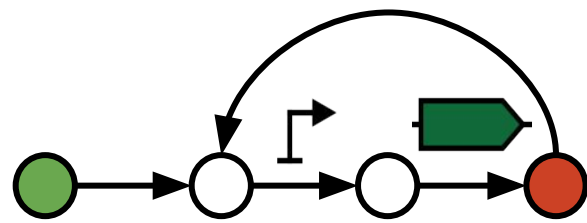


Changes to a Design Space Can Be Saved to its Version History



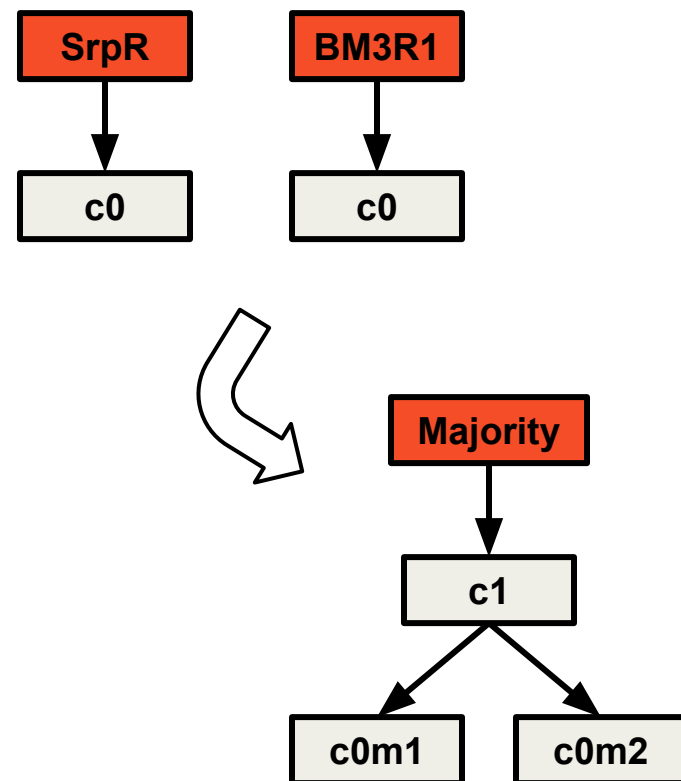
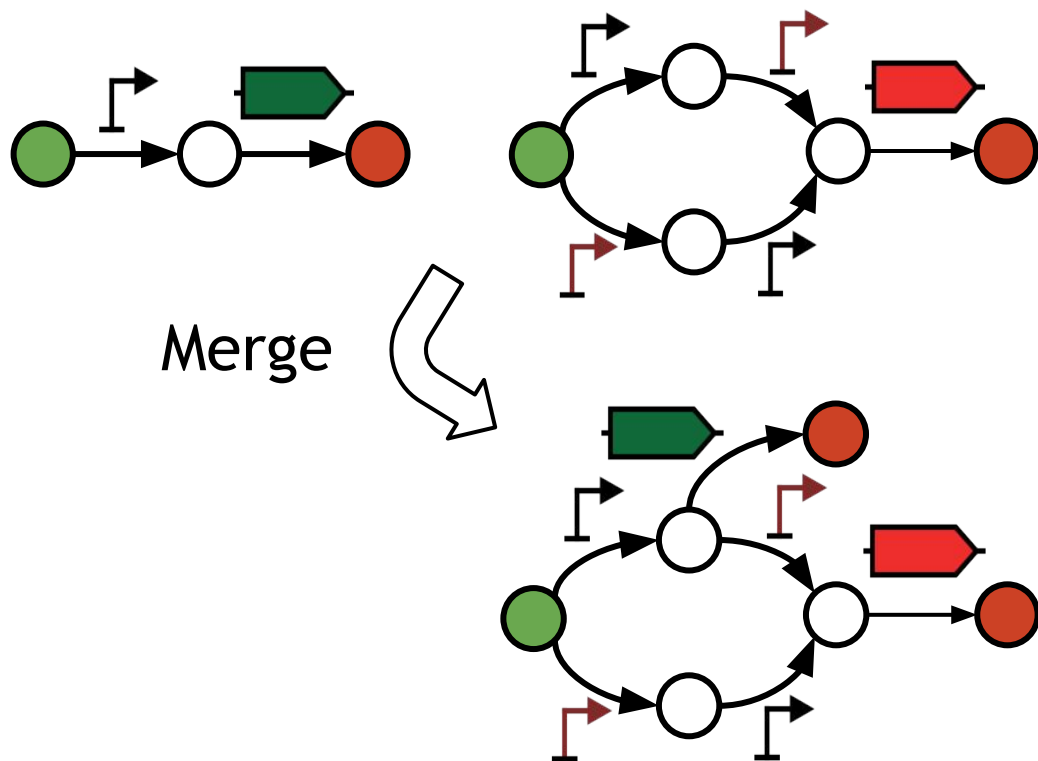


Changes to a Design Space Can Be Saved to its Version History



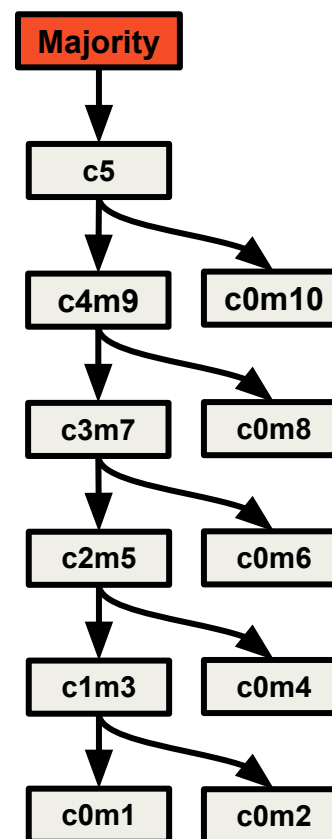
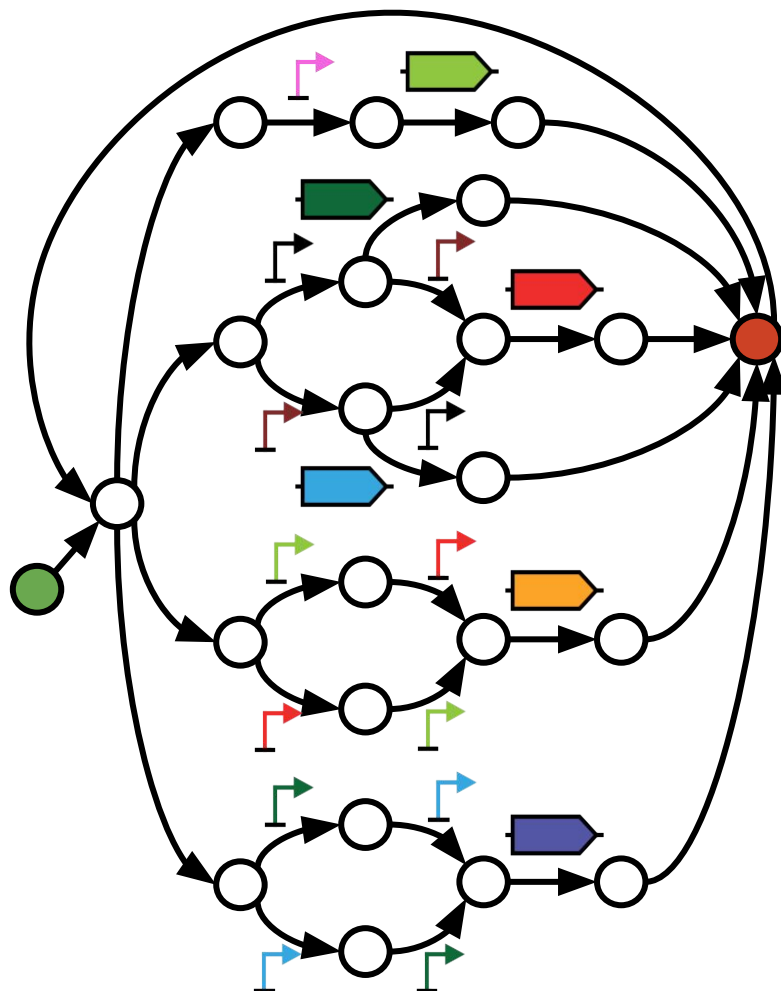


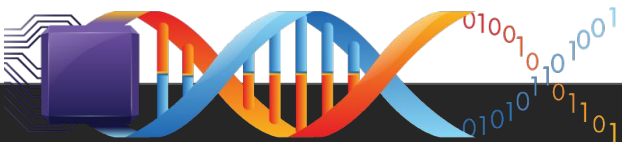
Combining Design Spaces in Knox Also Merges Version Histories



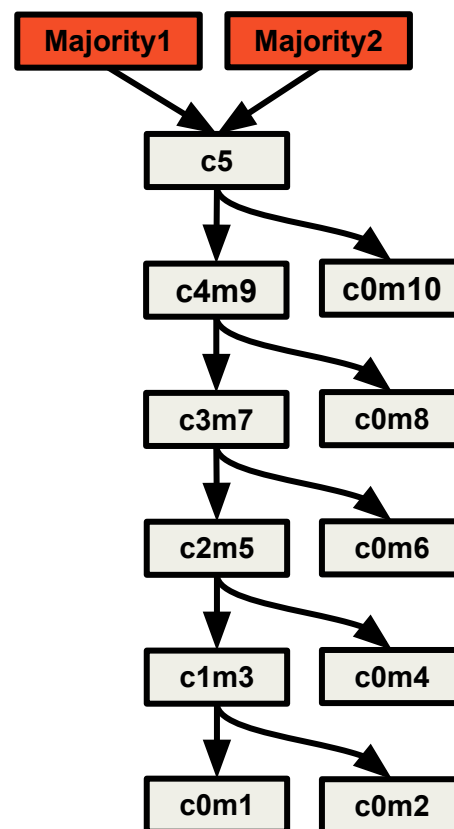
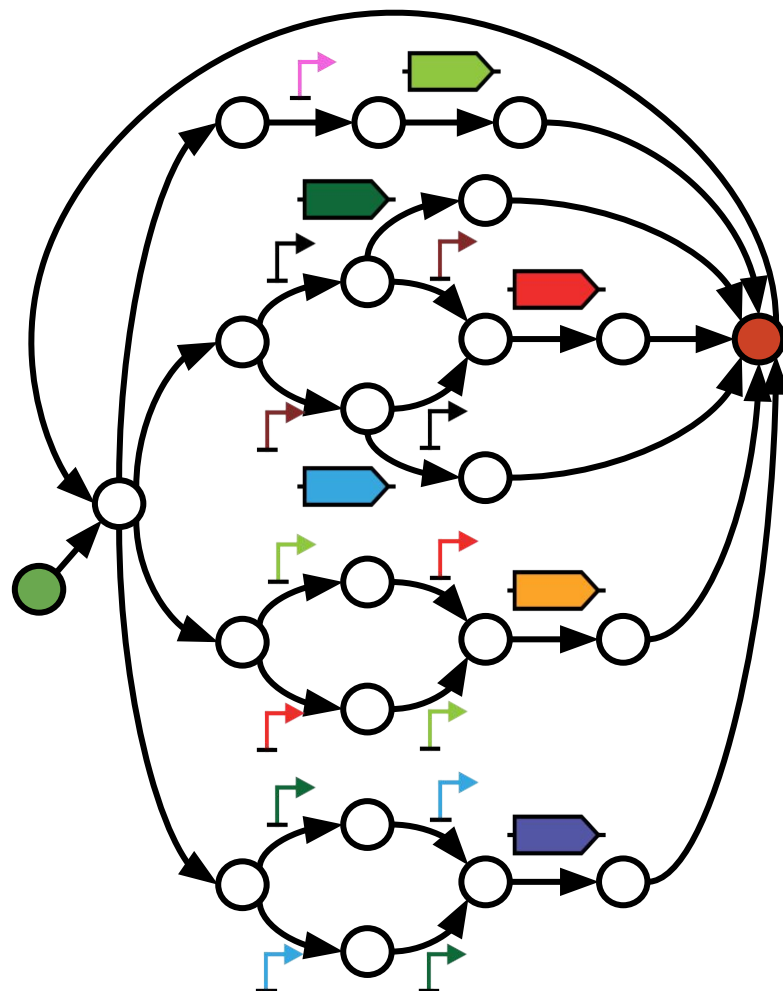


Version History for Majority Circuit



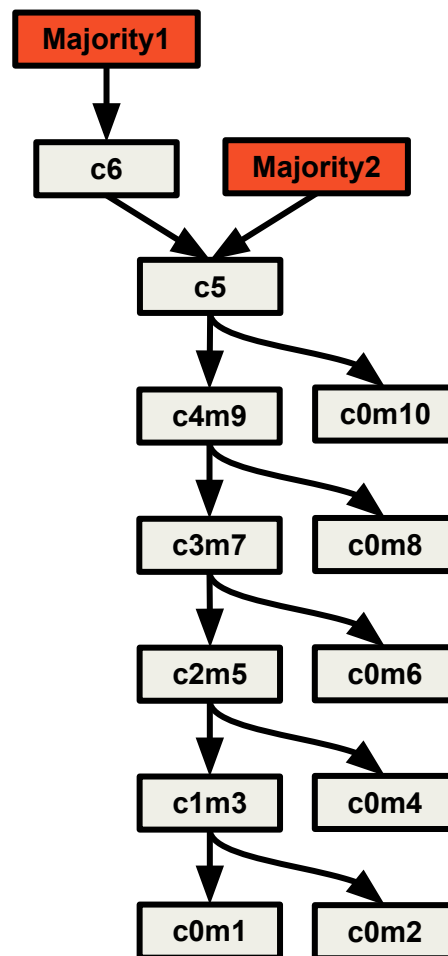
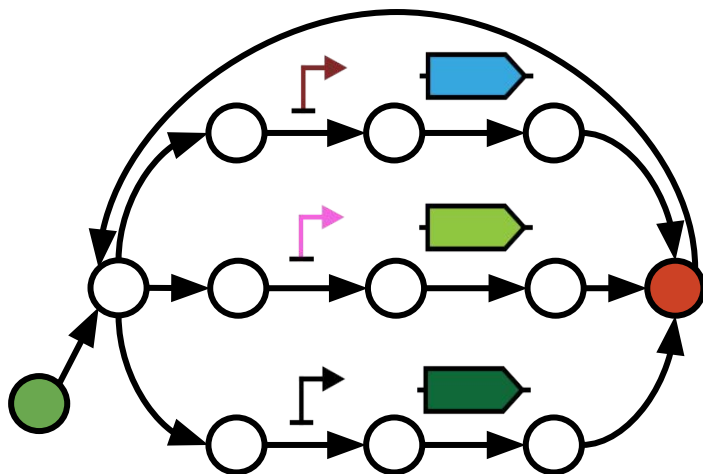


Branching a Design Space in Knox Creates Two Parallel Versions



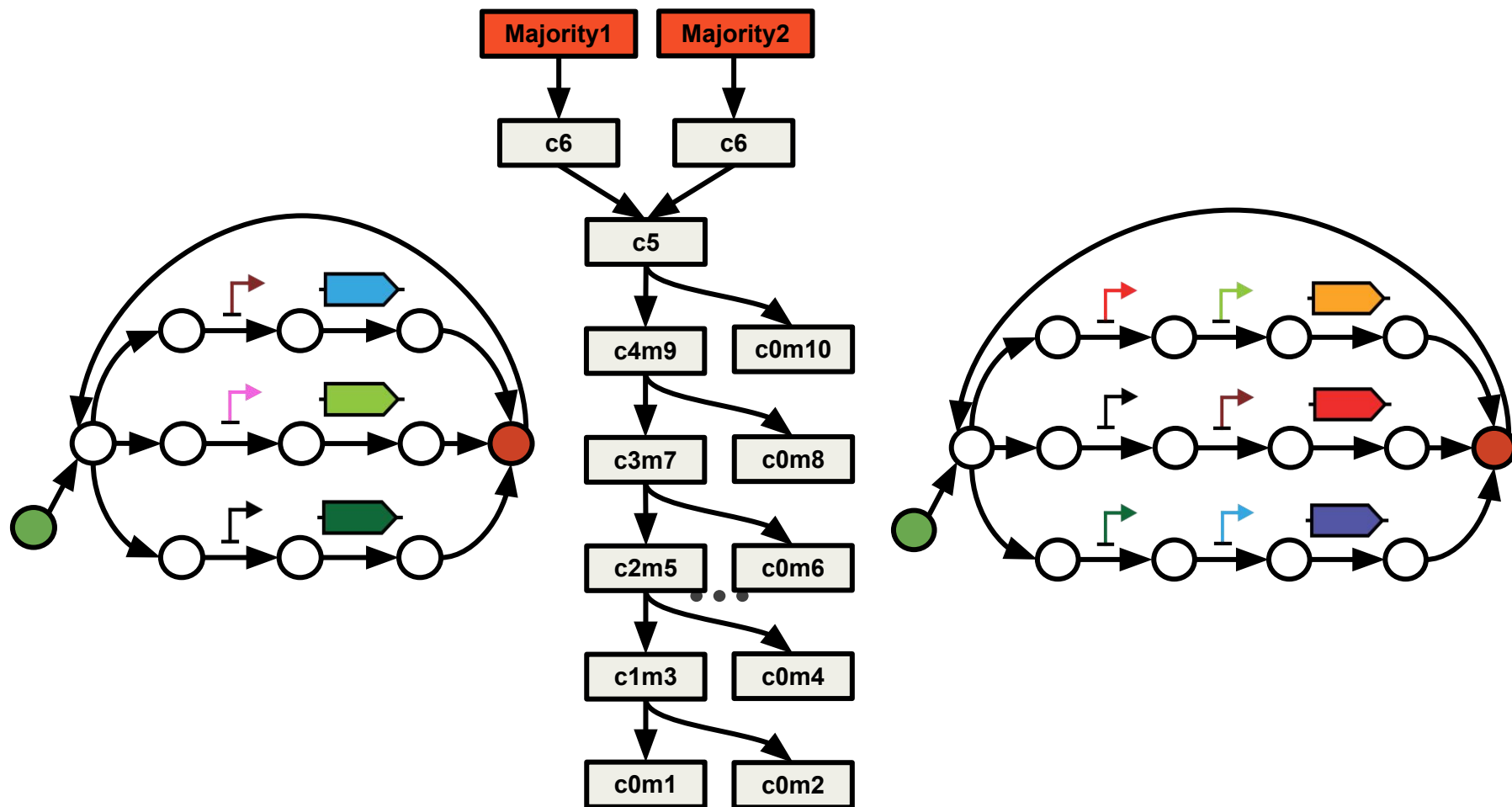


Further Refinements Can Be Saved to Separate Branches



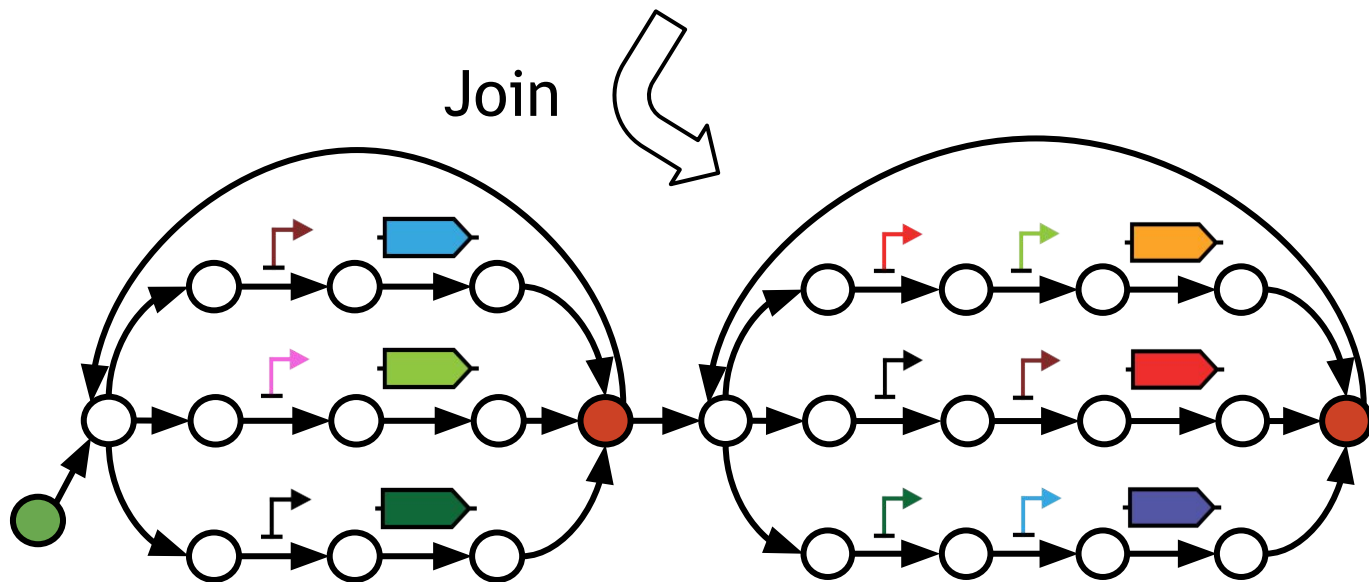
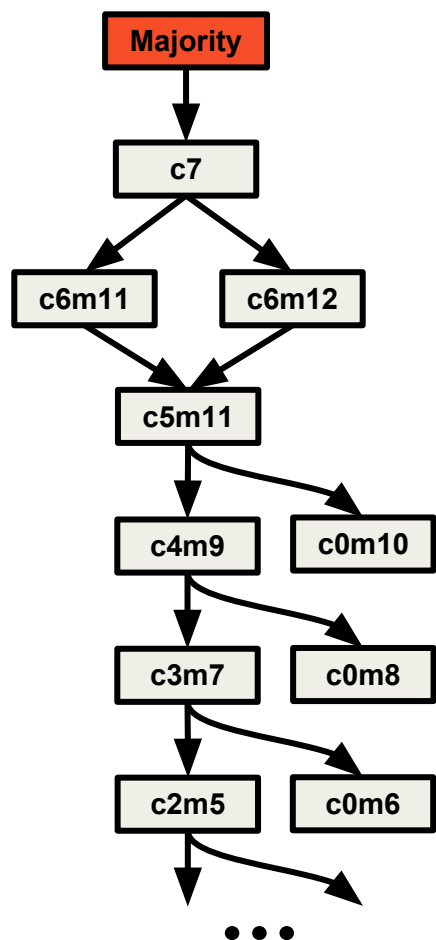


Further Refinements Can Be Saved to Separate Branches





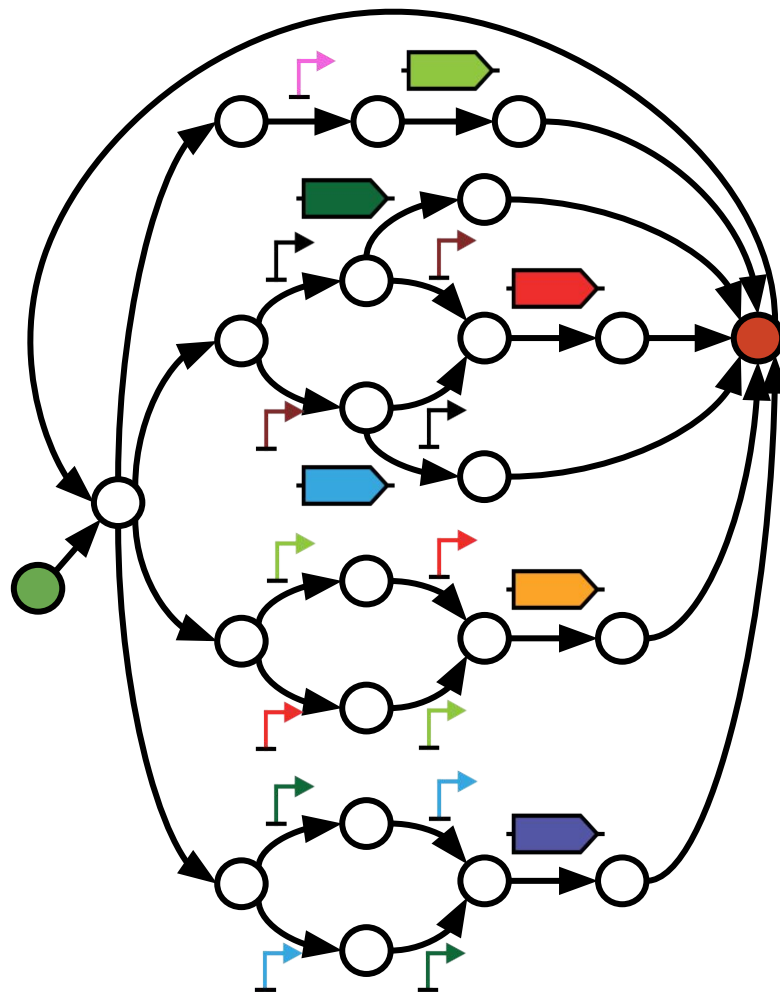
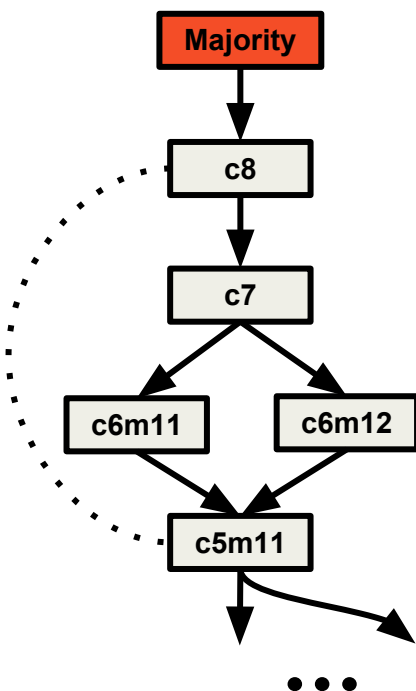
Branches Can Be Recombined Using Any Operator in Knox





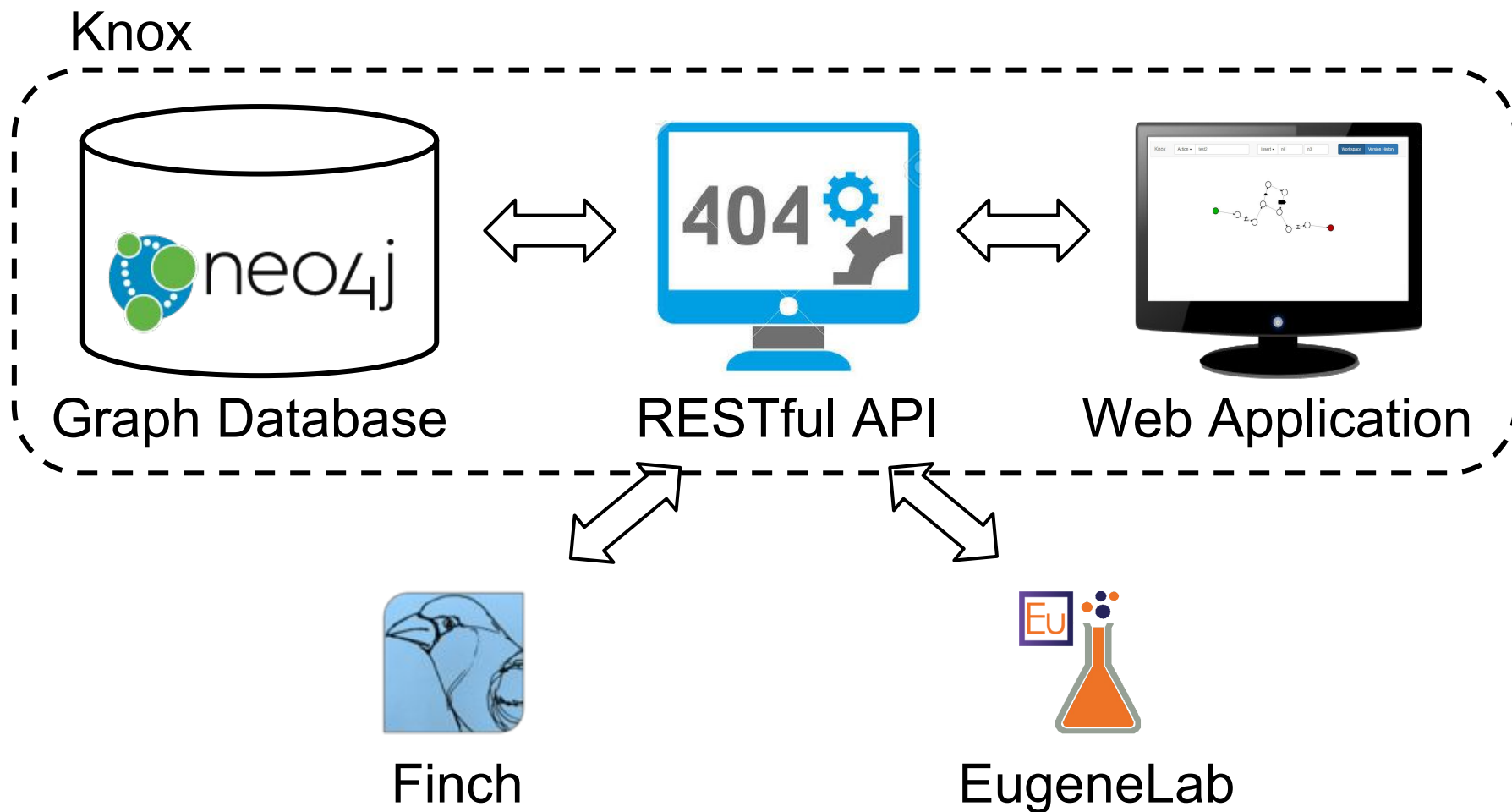
Previous Commits Can Be Revisited with the Revert Operator

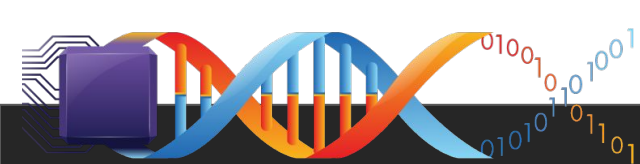
c8 is a copy
of c5m11





Knox Provides a Web-Accessible Platform to Store and Track Changes to Design Spaces





Finch vs. Eugene



Finch

- Variable design length
- More machine-comparable, -mergable



Eugene

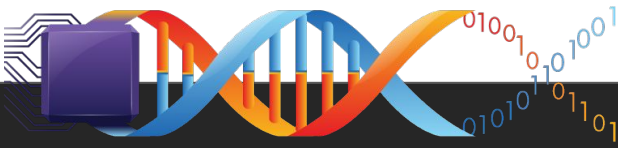
- Fixed design length
- More human-readable, -writable



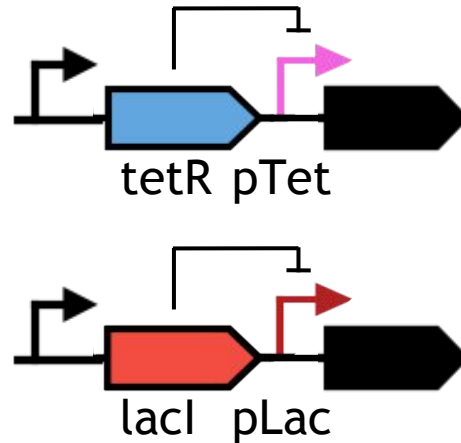
Eugene for Two Genes



Device toggle(Promoter, CDS, Promoter, CDS);



Constrained Eugene for Two Genes



Device toggle(Promoter, CDS, Promoter, CDS);

Rule toggleRules (ON toggle:

ALL_FORWARD AND

tetR BEFORE *pTet* AND

lacI BEFORE *pLac*

);



Compiling Eugene to Two Gene Design Space

Device toggle(Promoter, CDS, Promoter, CDS);

Rule toggleRules (ON toggle:

ALL_FORWARD AND

tetR BEFORE pTet AND

lacI BEFORE pLac

);



Compiling Eugene to Two Gene Design Space

Device toggle(Promoter, CDS, Promoter, CDS);

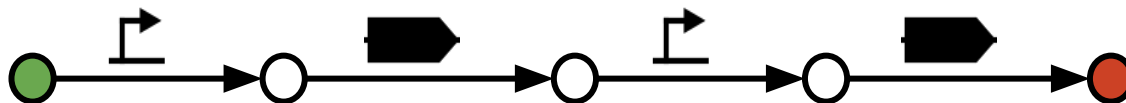
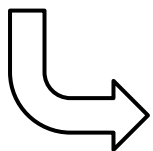
Rule toggleRules(ON toggle:

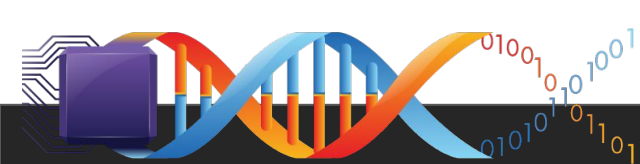
ALL_FORWARD AND

tetR BEFORE pTet AND

lacI BEFORE pLac

);





Compiling Eugene to Two Gene Design Space

Device toggle(Promoter, CDS, Promoter, CDS);

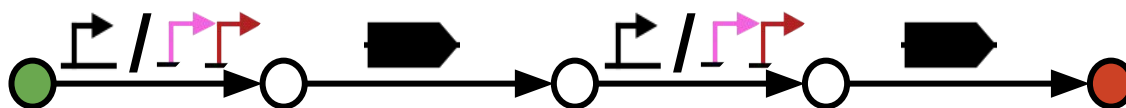
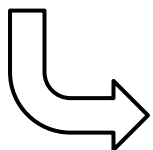
Rule toggleRules(ON toggle:

ALL_FORWARD AND

tetR BEFORE pTet AND

lacI BEFORE pLac

);





Compiling Eugene to Two Gene Design Space

Device toggle(Promoter, CDS, Promoter, CDS);

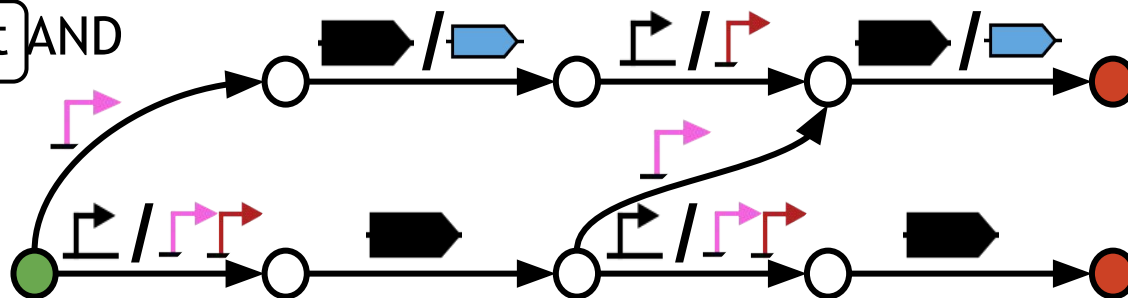
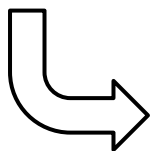
Rule toggleRules(ON toggle:

ALL_FORWARD AND

tetR BEFORE pTet AND

lacI BEFORE pLac

);





Compiling Eugene to Two Gene Design Space

Device toggle(Promoter, CDS, Promoter, CDS);

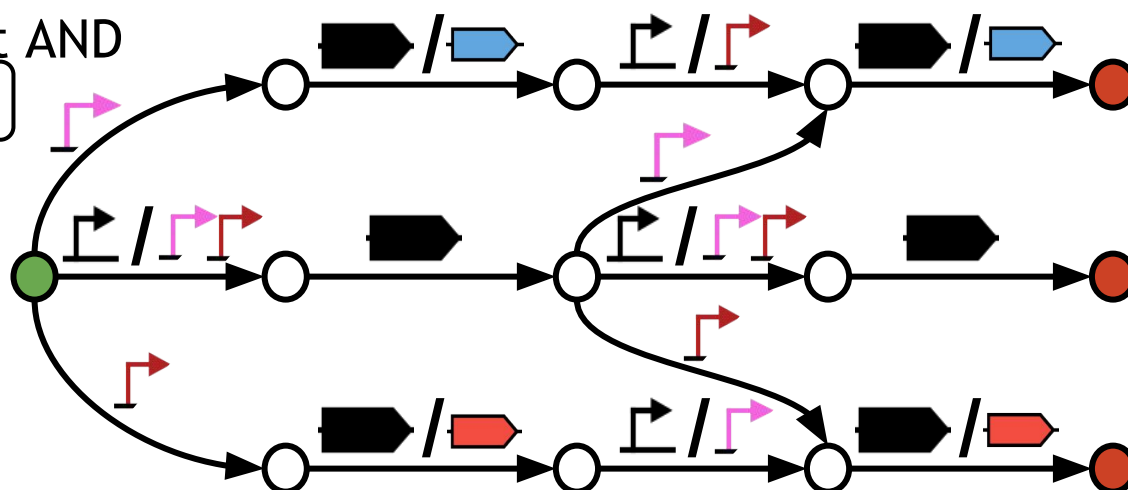
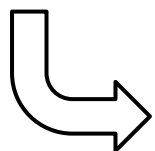
Rule toggleRules(ON toggle:

ALL_FORWARD AND

tetR BEFORE pTet AND

lacI BEFORE pLac

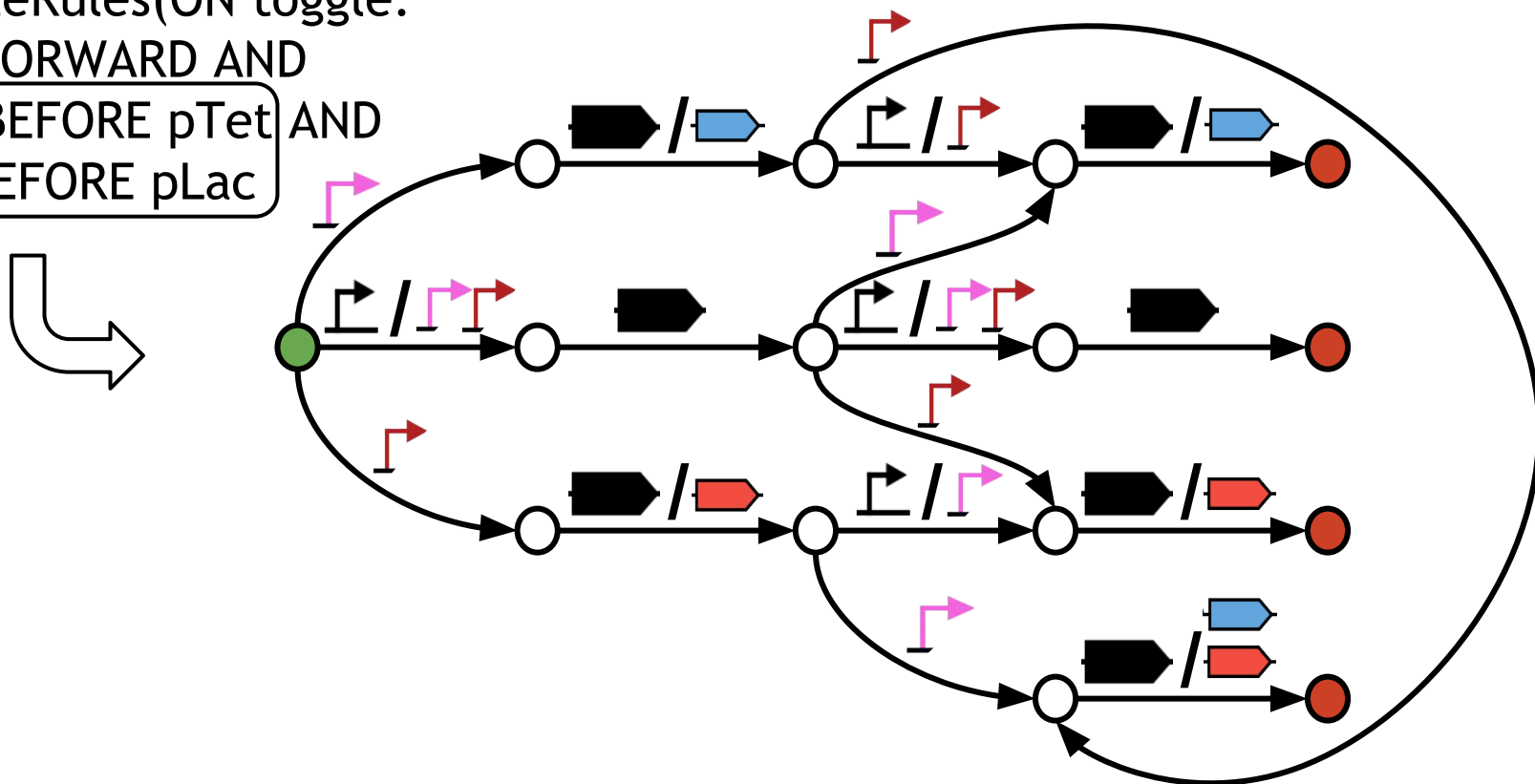
);





```
Device toggle(Promoter, CDS, Promoter, CDS);
```

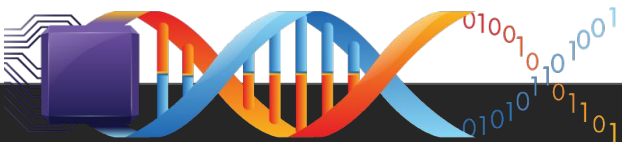
ALL FORWARD AND
tetR BEFORE pTet AND
lacI BEFORE pLac

$$);$$




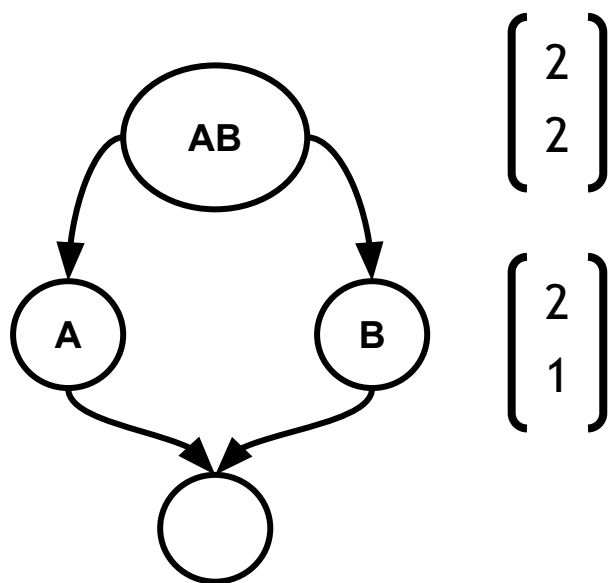
Finch Code for Two Gene Design Space

```
{  
    pTet THEN NOT tetR THEN {  
        { pLac THEN NOT lacI,tetR } OR  
        { NOT pLac THEN NOT tetR }  
    }  
} OR {  
    NOT pTet,pLac THEN cds THEN {  
        { pTet THEN NOT tetR } OR  
        { NOT pTet,pLac THEN cds } OR  
        { pLac THEN NOT lacI }  
    }  
} OR {  
    pLac THEN NOT lacI THEN {  
        { pTet THEN NOT lacI,tetR } OR  
        { NOT pTet THEN NOT lacI }  
    }  
}
```

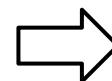
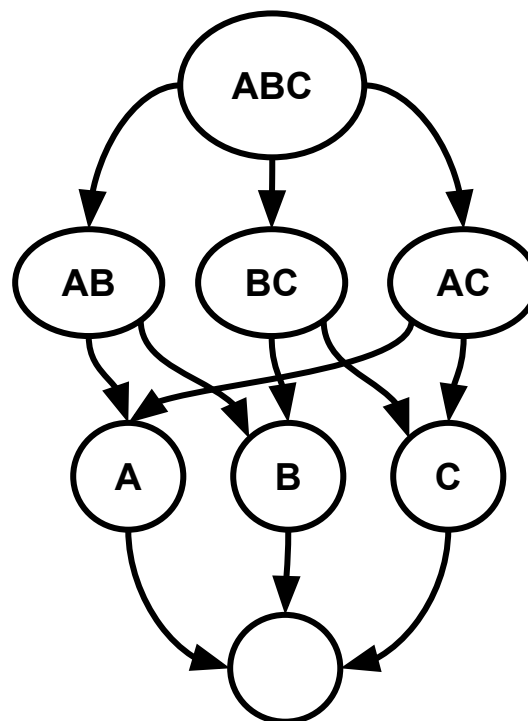
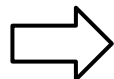


Scaling for Eugene Compilation

2^R where R is number of independent precedence rules



$$2^2 = 4$$



$$2^3 = 8$$



Acknowledgments



Prof. Douglas
Densmore

Prof. Swapnil
Bhatia

And the CIDAR Lab



D. Benjamin
Gordon



Prof. Christopher
Voigt

And the Broad Institute

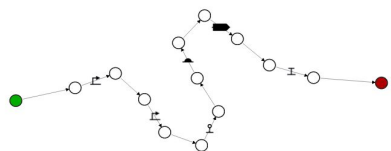
This research was funded by the DARPA Living
foundries award HR0011-15-C-0084.





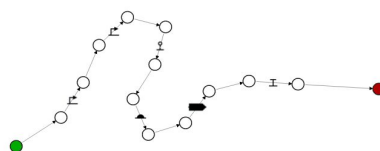
Thank You - Knox GUI in Action

Action ▾ test5 Insert ▾ Node ID 1 Node ID 2



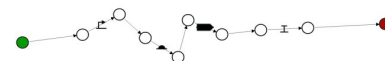
Join

Action ▾ test10 Insert ▾ Node ID 1 Node ID 2



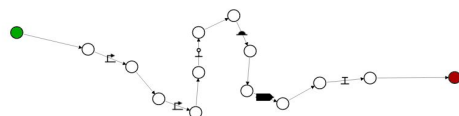
OR

Action ▾ test2 Insert ▾ n5 n1



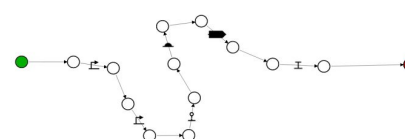
Insert

Action ▾ test11 Insert ▾ Node ID 1 Node ID 2



AND

Action ▾ test15 Insert ▾ Node ID 1 Node ID 2



Merge